LAUREA
UNIVERSITY OF APPLIED SCIENCES
Together we are stronger

Creating a modern and responsive customer tracking dashboard

Zuga-Divre, Liliana

2016 Leppävaara

Laurea University of Applied Sciences
Leppävaara

Creating a modern and responsive customer tracking dashboard

Liliana Zuga-Divre
Degree Programme in BIT
Bachelor's Thesis
November, 2016

Laurea University of Applied Sciences **Abstract**
Leppävaara
Degree Programme in BIT

Zuga-Divre, Liliana

Creating a modern and responsive customer tracking dashboard

Year 2016 Pages 72

Much of the success of many businesses is based on understanding the customer's interests and needs and working to improve the service or product offering based on this knowledge. For many big businesses this is relatively straightforward, as many tools are offered that can help them gather and analyse this information. For small and medium businesses, however,such an understanding may be much harder to achieve. Many of the tools and services do not fit into their available budget and might not bring in the results that would justify the expense. This thesis project was commissioned by 1UP Media, which, through their Lumi platform, offers a low cost, user-friendly, modular service for such businesses.

The thesis briefly introduces the company, the theoretical background and the technologies used in the implementation of the design for the Lumi platform people counting tool UI, such as Sass and Flexbox. Additionally, it explains the methodology, User Centred Design, and analyses the feedback received from live testing. The objective was to create a dashboard which would allow the data to be visualized, to ensure the understanding of the end user.

In this thesis project, the User Centered Design framework is used. The user-centered approach is one of the most popular frameworks to use when designing a new application or a website. It has a semi-cyclical approach, allowing a close interaction with the client for constant improvement before a pilot is conducted. By maintaining this closeness with the client, it is possible to reach the most satisfactory and most user-friendly end product, therefore the overall quality can be ensured.

As the result of this thesis, a pilot version of the designed application was completed and prepared for further improvements based on any additional user feedback. This fulfills the requirements set at the beginning of the project and the additional requirements set after the first live testing. These requirements mainly focused on making the application modern, responsive and easy to modify were the need for rebranding to arise. These were achieved through the research of the appropriate literature and application of the gained knowledge from this theory to the product, improving the design as issues were brought up in the first live test. As the application improves and expands, further changes may be required, causing the design to change to accommodate the needs.

Keywords: User Centered Design, live testing, designing

Laurea-ammattikorkeakoulu **Tiivistelmä**
Unit
Programme

Zuga-Divre, Liliana

Modernin ja responsiivisen hallintatyökalun visuaalinen suunnittelu asiakaskäyttäytymisen mittaamiseen ja suunnittelemiseen

Vuosi 2016 Sivumäärä 72

Asiakastarpeen täyttyminen on ensisijaisen tärkeää liiketoiminnan kannalta. Asiakastarpeen ymmärtäminen auttaa yrityksiä parantamaan tuotteitaan ja palveluitaan. Asiakaskäyttäytymiseen liittyvän tiedon keräämiseen ja analysoimiseen on olemassa työkaluja, mutta ne ovat suunnattu suurille yrityksille. Pienille ja keskikokoisille (PK-yrityksille) asiakastiedon kerääminen on huomattavasti hankalampaa. Olemassa olevat työkalut ovat joko liian kalliita, tai eivät tuota käyttökelpoista tietoa PK-yrityksille. Tämä on ongelma, johon opinnäytetyön asiakas 1UP Media OY tarjoaa ratkaisun; Lumi-alustan. Lumi-alusta on edullinen, käyttäjäystävällinen ja joustava palvelu PK-yrityksille.

Tämä opinnäytetyö koostuu seuraavista osioista; aluksi esitellään asiakas ja käydään aläpi teoreettininen tausta ja visuaalisessa suunnittelussa käytetyt teknologiat, kuten Sass ja Flexbox. Tämän jälkeen opinnäytetyö käsittelee käyttäjälähtöistä suunnittelua – metodologiaa, johon opinnäytetyön visuaalinen suunnittelu perustuu. Lopuksi analysoidaan testauksessa mukana olleiden asiakkaiden palautetta. Opinnäytetyön tavoitteeksi on asetettu käyttäjäystävällisen hallintapaneelin suunnittelu, jonka avulla loppukäyttäjän on mahdollista visualisoida ja tulkita asiakastietoa.

Tämä opinnäytetyö pohjautuu käyttäjälähtöinen suunnittelun viitekehykseen. Tämä on yksi suosituimmista lähestymistavoista uusia verkkosivustoja tai sovelluksia suunniteltaessa. Käyttäjälähtöisen suunnittelun viitekehys perustuu jatkuvaan yhteistyöhön asiakkaan kanssa ja se mahdollistaa jatkuvan kehityksen jo ennen ensimmäistä pilottiversiota.

Lopputuloksena on pilottiversio hallintapaneelista ja tuote on valmis muokattavaksi asiakkaan palautteen perusteella. Tuote täyttää projektin alussa asetetut vaatimukset, sekä lisävaatimukset, jotka asetettiin ensimmäisen live-testauksen jälkeen. Vaatimukset keskittyivät pääasiassa tuotteen responsiivisuuteen, nykiaikaiseen ulkoasuun, sekä helppoon muokattavuuteen. Vaatimukset täytettiin tukeutumalla teoreettiseen viitekehykseen ja jatkuvaan asiakaspalautteeseen.

Avainsanat: Käyttäjälähtöinen suunnittelu, Live testaus, käyttöliittymä, käyttäjäkokemus

Table of contents

Terms and abbreviations

2D	2nd dimension, a flat surface image
3D	3rd dimension, an image with width, depth and height
CSS	Style Sheet language
FAQ	Frequently Asked Questions
Google Drive	Cloud storage and file backup provided by Google Inc.
GNU	Software that anyone can download, modify and distribute.
Head	Tag element that includes a title for the document, scripts, styles, etc.
HTML	Hypertext Markup Language
HTML Tag	Markup language code for Web page formatting. Defined by the < and > characters.
Module	A set of independent units that can be used in a more complex structure.
in	Inch. Measurement of 2.54 cm
Prototype	A first or early example that is used as a model for what comes later
Px	Pixels. Measurement unit
UI	User Interface
UML	Unified Modeling Language
UX	User Experience

1 Introduction

The past few decades have seen a big change in the way that the information is handled and a lot of fields have completely abolished their original methods in favour of the advances of the computed ones. Nowadays, it's a big industry of its' own, the Information Technology industry, and the newest services are built around it, making our lives increasingly more computerised. With this, it's become a lot more important that the offered applications are visually appealing and user friendly. Any web-based product is quickly judged by the first impression. If the product looks too complicated or too narrow for their needs, the user might not be willing to try using it.

1Up Media Oy is in the process of creating a new platform and it's important to them that the best web design practices are applied in the process. As a small, new company, it's essential to make a breakout into the market, making it imperative that the first product is successful, marketable, user friendly and that it fulfills some necessary function or solves an important problem.

Another crucial idea for the Lumi platform is the orientation for small and medium businesses, which usually are unable to afford similar services. With this focus, the company wants to create a light and fairly low-priced service that would allow such businesses have the same advantages as the bigger businesses that can invest more and may afford the current similar products in the market. By ensuring that the visualised data is clear and easy to analyse as well as generally ensuring that the application is not overly complex whilst still offering a wide range of necessary functions, it can affirm that the product will succeed in the market.

This paper will cover the creation of a design and implementation process of a user interface for visualising the collected data to the end user and allowing a clear and effortless analysing of it. It's divided in 8 main sections. The introduction part, describing the project, its' scope and limitations, and introducing the company. The methodology part, describing the frameworks used in the research part of the project. The knowledge base part covers the theories researched to conduct the project work. The tools and languages and modules parts cover the markup and extension languages, as well as the modules used to implement the project. The implementation process describes how the aforementioned knowledge base and technologies were used to achieve the project goal. Finally, the evaluation and conclusion parts evaluate the project at the final stages of the thesis.

1.1 1UP Media Oy

1UP Media Oy is a Technology startup company located in Leppävaara, Finland. It specialises in software development and provides tailored solutions and industrial software. It was founded by Tom Forsman and recently the company has begun cooperation with rail and transit technology companies. Currently 1UP Media Oy has 3 employees and several interns, working on the solutions with a goal-oriented mindset and developing the current project – Lumi.

1.2 Project background

The project currently developed is Lumi; a cloud-based software for businesses. This platform consists of several services, one of them being Lumi people tracking module.

Lumi people tracking is a unique technology for businesses that allows tracking the flow of customers, collecting the data in non-media format. This allows the businesses to analyse their customer preferences on when to use their services the most and which parts of the service are the most utilised. Using this data, the business owners can reallocate staff upon need and to observe the possible maintenance requirements in the most frequently used service locations.

At the time of this research project start, some core functionality had already been created, but only some ideas were considered for the actual design of the application. The overall project consists of several parts: The user interface accessed directly by the user and the devices, the data collection software, the API which is out of the user's direct reach and serves as the foundation of modules and their functionality, and database where the data is securely stored.

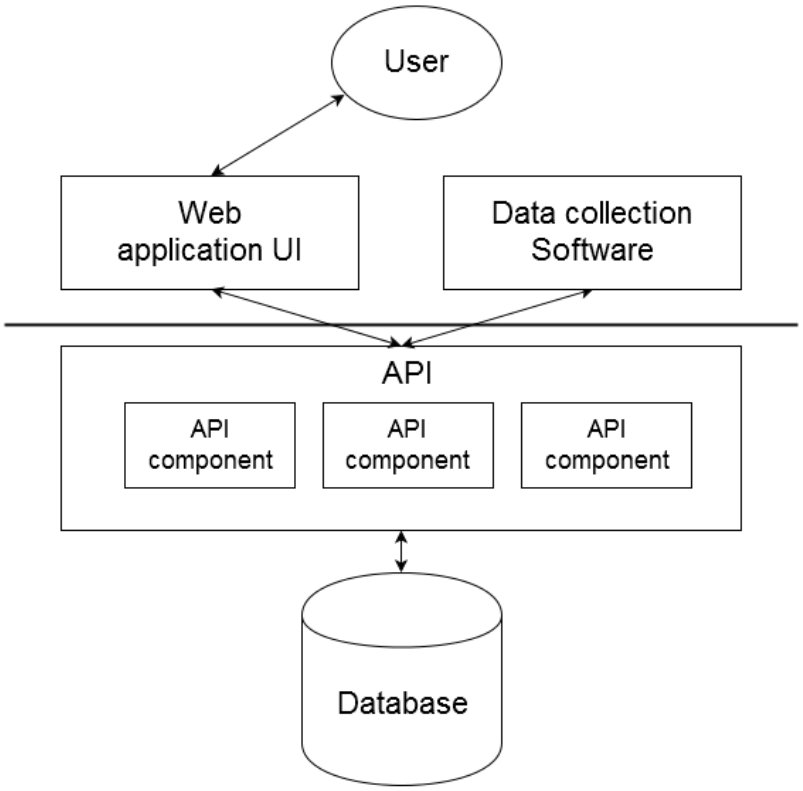


Figure 1: Lumi platform general architectural design

1.3 Objectives

The main objectives of this thesis and the major requirements received from the company are the following:

- Create a modern UI for the platform's reporter and configuration side,
- Create wireframe, prototype and mockups,
- Ensure the responsiveness of the design,
- Implement the approved designs.

The minor requirements are:

- Create different templates and colour palettes for end user to switch between,
- Help with creating different types of branding possibilities within the platform.

1.4 Project scope and limitations

This thesis focuses on the design process of the application, and the ways to make it most marketable and user friendly. It doesn't cover the functionality of the application and only briefly explains the technology used as the foundation for the web application. In the

development, the budget limitations affected the used technologies for the mockup design, but didn't affect the implementation. The project is currently incomplete and a subject of any possible changes in the near future, therefore no market numbers can be confirmed.

2 Methodology

In this section the methods used in the research and development process of the project are described. It covers the framework of the development – User-Centred design framework, and the methods of research, such as benchmarking, wireframing and mockups. Additionally, user testing was conducted as an evaluation method of the first design version.

2.1 Development process

The framework used in the development of the project was User-Centred design (UCD) framework. As quoted on W3C website from ISO 13407, "Human-centred design is an approach to interactive system development that focuses specifically on making systems usable. It is a multi-disciplinary activity." (W3C 2008)

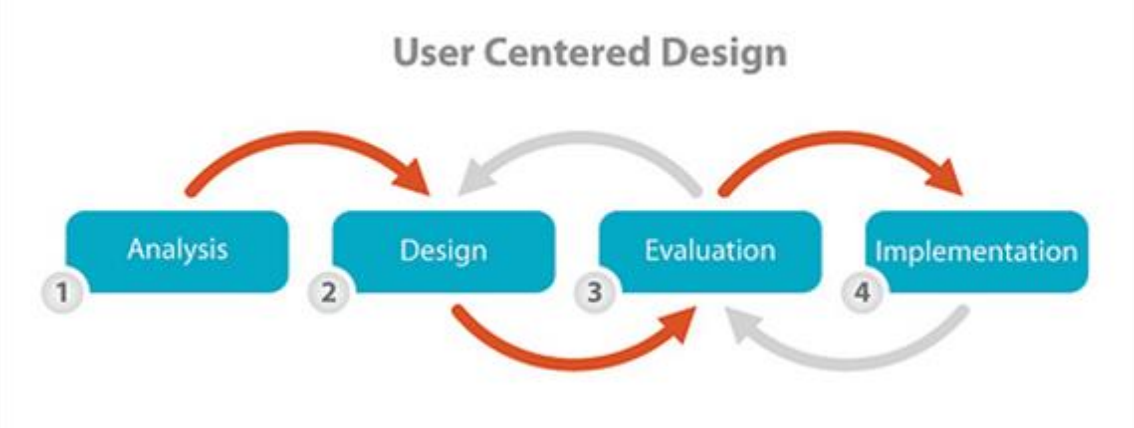


Figure 2: User Centred Design framework (Weevers 2012)

The initial process, as shown in Figure 2, goes through four phases before reaching the deployment: Analysis, Design, Evaluation and Implementation. These phases interact among themselves in back and forth approach, where each phase is not necessarily final and the process can return to the previous phase for improvements and changes.

During the Analysis phase, a large set of information is gathered. Firstly, the designer must gather the information from the customer; their vision, the goals of the final product, the objectives and constraints. Once these have been gathered, User/Audience analysis, Task/Purpose analysis, Information architecture analysis and workflow analysis must be

conducted to understand how the customer's vision and requirements align with the needs of the market that they are trying to approach.

After Analysis has been finished, the Design phase can be begun, where, with the use of wireframes and mockups, the ideas are presented to the customer. This phase is closely related with the next phase that is Evaluation. By understanding how the design works and by using the customer evaluation of it, it can therefore be adapted to the customer needs but also ensured that it follows the needs of the end users of the target group.

Along with Design and Evaluation, the Implementation phase can iterate with the the other two phases in the process. When implemented, the design is evaluated and, if necessary, re-designed to achieve the customer expectations using the best practices for the design. Once the implementation phase is set as finalised, the final product can be deployed.

If the product has to go through any additional changes, the whole process is restarted.

2.2 Benchmarking

Benchmarking is "a measurement of the quality of an organization's policies, products, programs, strategies, etc., and their comparison with standard measurements, or similar measurements of its peers." and one of the objectives for it is "to analyze how other organizations achieve their high performance levels".

When designing a web application, it's an important step to take, allowing the designer to avoid possibly obvious mistakes and not forget elements that are clearly needed. Many examples can be found online, making the benchmarking process quicker, though this process still accredits to a few day to few week time in the project process. During this time, a list of good and bad practices can be created.

The web design benchmarking process follows these steps:

- Identifying goals,
- Identifying website's components,
- Identifying website's weaknesses and prioritising,
- Choosing benchmarking websites,
- The benchmarking process itself,
- Creation of highlights report.

(Esparza 2013)

By following these process steps, it's possible to achieve the final documentation which would therefore allow the design process to continue.

2.3 Wireframing

Wireframing is one of the first design process steps in which the first concepts are drawn either on paper or created digitally, but only includes the basic layout which implies the functionality without actually including it. Additionally, it presents the structure and content of a website. (Webopedia n.d.)

Wireframes have their advantages and disadvantages. The advantages are that the wireframes are easy to create and alter, allowing to go on with the project with more confidence once the wireframes have gone through the approval of the customer. The disadvantages are that wireframes are very plain and might not be easily understood functionality-wise by the customers. Additionally, once actual design is implemented, it might not translate well with the ideas originally in the wireframe, therefore requiring more work to make sure that the content and structure of the website work as intended. (ExperienceUX n.d.)

2.4 Mockups

Mockup is defined as ''A full-sized model of something that is used for studying, testing, or showing its features.''. (Merriam-Webster n.d.) It's an essential part in the design process, where, according to the framework and with the use of graphical editing tools such as Photoshop or Illustrator, the visual aspects of the website are created and presented to the customer.

One of the main goals for this process is a sample of the website in which the design would display clearly the layout of the content. With this sample, it's easier to understand how the layout works in a more realistic situation than what the purely conceptual level of wireframing can display. Additionally, it works as a tool for feedback gathering. (Treder 2016)

2.5 User testing

User testing, also known as Usability testing, is the process in which a potential user attempts to use the prototype of the final product. In this step, it can be determined if the user experience is intuitive and if the website is understandable without the need of extra guidance. (Usability Testing n.d.)

There are several methods for user testing, but the one used in this project is one-on-one interview. By creating a discussion guide – a set of premade questions that are asked in a certain order, - the test user is interviewed about the experiences of using the website. Additional questions and user observation are also a potential part of the approach that can be utilised to get the best results without a bias. (One-on-One Interviews n.d.)

3 Knowledge base

This section covers the concepts used in the project. Firstly, the general design principles and psychology are covered, followed by more specific styles and concepts that must be applied in the designing process.

3.1 Design principles

When creating any web design, a set of established principles should be followed to ensure that the created UI provides the best quality UX. Due to the history and practices of web design changing quickly, it may be difficult to follow everything, but a list of principles always remain.

These principles can be separated into four main sections:

- Axis,
- Symmetry,
- Hierarchy,
- Rhythm;

Axis is an imaginary line used for organising the elements of a web UI. Upon experiencing this invisible line, the user doesn't have to spend a lot of time looking for the desired content, instead they must simply follow the linear arrangement of the items displayed. Following this approach, the UX is improved considerably and it helps the design to become a lot more intuitive.

This line also serves as a tool for better designing the websites and applications. Even when an element might be missing or displayed at a different distance from the others, the axis principle reinforces the understanding that the element remains the same onwards in the design. In turn, this also shows the effect of this principle in understanding the direction of the layout, once again improving the intuitivity of the UI.

Symmetry is a principle that must be included into the design for the sake of the user comfort in viewing the content. By creating symmetric elements, the content becomes more balanced and easier to follow. It doesn't create the anxiety of trying not to miss something important and maintains the attention of the users. Although asymmetric design is not uncommon, it's not fit for a design which should display all the content equally visibly.

Hierarchy allows the content to be arranged in the manner from the most to least important, using size, shapes and arrangement within the page. Bigger elements will always seem more important. The differently shaped elements will draw the user's attention in first. Finally, an element positioned at the top of another element will make it stand out better, and therefore make it seem more important to view first.

Rhythm is important to create the sense of familiarity within the design by using same or similar elements regularly. Using this approach any new element will stand out with the interruption of the familiar rhythm. This principle also helps with making the user experience more pleasant; when the elements have become familiar, the viewing on the contents becomes considerably more enjoyable for the user. There is no need to re-learn the flow of the UI where the content elements are mainly the same in form and importance. Bringing attention to content can be done by utilising different design and breaks, as these methods indicate a change in importance. (Mandelbaum 2014)

3.2 The Gestalt principles

Gestalt is a psychological term for "unified whole", and as a principle can be used for web design. It bases on the human need to organise elements based on certain rules, such as those listed:

- Similarity,
- Continuation,
- Closure,
- Proximity,
- Figure and ground;

and while the elements may not always instantly look what one could consider organised, these principles are used subconsciously by the mind to group and organise. Knowing these subconscious tendencies can help directing the user's attention to or from certain parts of the design and therefore - content.

Using similarity principle and therefore anomalies - elements that differ, are dissimilar, from the rest of the content - user's attention can be drawn towards some special announcements and offers. Meanwhile the regular, less urgent content can blend in with the rest of the UI by simply maintaining the similarity of the design. Another way to direct attention of the user is by using continuation. Using the shape of one element, or more, shift the attention to another element besides the first one. It can be achieved by directive shapes and curves.

These aforementioned elements don't mean that all the elements must be fully finished to smallest detail. Where the element is not finished, the appropriate display of the current elements can cause the mind to fill in the remaining and see the whole image without the need to make it complicated. It must be kept in mind, however, that continuation occurs only when the elements are not spread too far apart. This is also a thing that is important with every element in the design; grouping elements is a direct way to make the user see them as similar and therefore acknowledge them as of similar importance.

Lastly, the human eye and mind creates a figure (object) and ground (background) when viewing a certain type of group of visual elements. This can, and is, used in web design as many cases put a ground element that may be very similar to the figure element, but still is perceived as elements placed one on top of the other instead of side-by-side. (Spokanefalls n.d.)

3.3 Creating modern web application designs

Creating design for a web application may not mean that the application will surely follow the advancements in the current web design trends. This means that a certain amount of research must be put into understanding these trends prior designing. In this section, several of the currently most popular styles will be described to better understand the theory of what currently makes a modern design and what is by now an outdated trend.

3.3.1 Skeuomorphism

Skeuomorphism is a design principle which uses real world approach to its' elements. By adding decorative effects, the elements are made to stand out, such as the keys on the earlier iOS keyboards, where the shadow beneath each key was applied to make it look more like a physical keyboard. Originally a very popular design approach for Apple, it has since become outdated with the newer user generations becoming less familiar with some of the objects and functions in physical format, such as calendars.

Though not completely abolished, there are a number of elements that have been almost completely discontinued or only partially utilised in the more modern designs. Some of these would be styling the buttons with a gradient, such as that seen in Figure 3, and using strong borders around them which make the edges rougher and any change feel like the button is actually being physically pushed into the page layout. (Skeuomorphism n.d.)

Figure 3: The difference between skeuomorphism and flat design buttons (Greever 2013)

3.3.2 Flat design

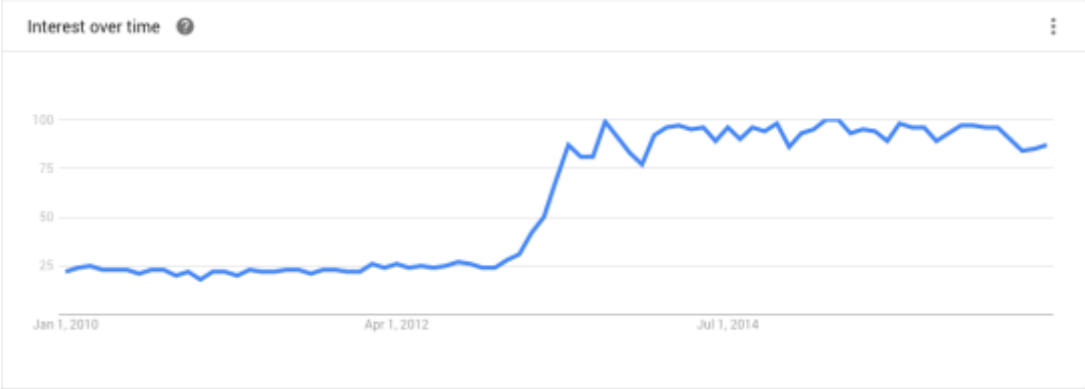


Figure 4: Google trends for Flat design 1.1.2010 - 27.09.2016

Flat design has been the trending style for the past few years, rising to fame at the end of 2012 with the introduction of Microsoft's Windows 8 platform, and, like the name suggests, it has a flat approach to it. The elements are not made to stand out with the use of 3D effects like it's done in Skeuomorphism and instead uses 2D approach and strong contrasts to create an understanding in the user. Flat design doesn't enclose too many rules, which serves as another reason for its popularity.

Some of the ways that Flat design achieves its' characteristic appearance are:

- Bright but muted colours,
- Bold typography,
- Simple UI elements.

As listed previously, one of the characteristics of Flat design is the use of bright colours that create a contrast. Unlike the first impression, however, the colours are not flashy and instead maintain a muted value. This is to avoid the possibility of the design becoming straining for the eyes when looking at it and therefore making it less user friendly. Some of the colour examples can be seen in the Figure 5.

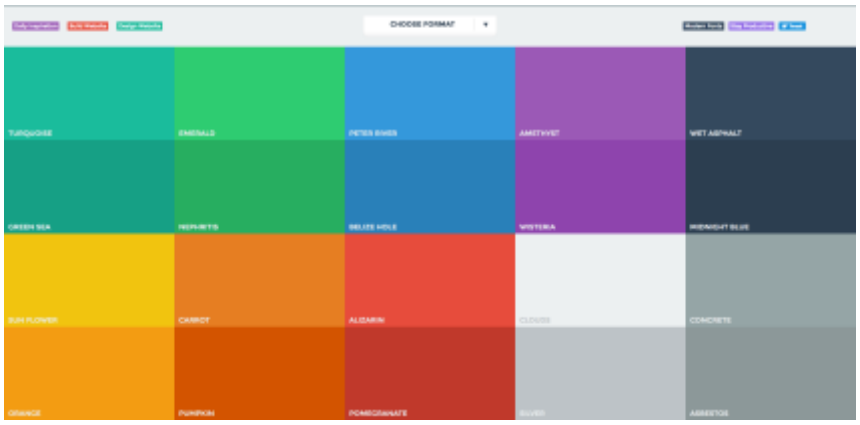


Figure 5: Most commonly used colour list (FlatUIcolors n.d.)

Of course, there is no true limitation to the colours used and even the same colours as in figure 4 can easily be altered. They can be made lighter or darker with the use of tints and shades, depending on the needs of the design. It will remain Flat design as long as it follows most of the Flat design principles.

Another mentioned characteristic is the use of simple, bold typography. Flat design largely abolishes the usage of cursive-like fonts and instead embraces the use of fonts that are crisp and easy to read. Such as in the Illustration 1, fonts like Robotic will be used more frequently than those similar to Savoye LET.

Caption 1: Flat design refrains from using intricate fonts

As one of the most characteristic features of Flat design, the use of simple UI elements is the most notable. As mentioned earlier, Flat design uses 2D elements instead of 3D, abolishing the use of shadows and gradients for the elements. This may sometimes make the UX more tedious, which means that more attention in design must be focused on ensuring that the functional elements stand out accordingly, by using colour changes, contrast and shape

changes as well as direct implications of functionality such as the text displayed on the buttons and similar elements. (Cousins 2015)

Due to the style's flexibility and simplicity as well as the overall popularity at the current time, it was chosen as the main style present in the application.

3.3.3 Material design

Material design was created by Google as an improved design of the increasingly popular Flat design. Unlike Flat design, where all the elements are completely flat and the difference of those is created by colours and contrast, material design applies the use of such attributes as shadows and elevation. By using the 3D principles of the three axes (x, y and z), it arranges the elements in the UI so they would be easily distinguished as separate and which of those are functional and which – static. It's commonly used in all Google products and Android operating systems since the release of Android Lollipop 5.0 in 2014.

While Flat design is largely a flexible design with many takes on it, Material design follows the standards created and frequently updated by Google. The guidelines allow the design to maintain its identity and ensures that it's easy to understand and use for the designers and developers.

Material is characterised by several different aspects. Firstly, it's solid, meaning that the material elements cannot pass through one another and they can't occupy the same space. Just like several pieces of paper, it can place the items side by side, it can stack them in z-axis, but they will never have the same location in the dimensions. Secondly, the content and events of the material are bound by their container elements. This means that the content can occupy any space in the material, but it will never leave its' bounds and the event won't be activated in any other material aside from the containing one. Additionally, the shadows that the material creates are based on real-life approach of elevated object shadows, depicting the distance between the elements.

There are certain characteristics of the material transformation and movement that are specific to Material design. Since material is solid, it doesn't bend or fold, instead it transforms along the x, y and z axes, changing the shape and size along these axes. It can also be manipulated to generate and destroy material and, by putting elements together and taking them apart, the material can be changed without unnecessary complication. This is due to the fact that material can ''heal'' to form a single, solid sheet again.

One of the most distinct things about material is that a majority of the z-axis movement is caused by the interaction of the user. By hovering over, grabbing and moving the elements, they firstly change the location in the z-axis before being manipulated further by the interaction.

Additionally, Google offers commonly updated icon list that can be specifically used in Material design. These icons are flat, so the use doesn't just stop at Material design and can be used across different styles. Though limited, it provides the most commonly necessary icons and the unified style allows the design to maintain the continuity. (Google 2014)

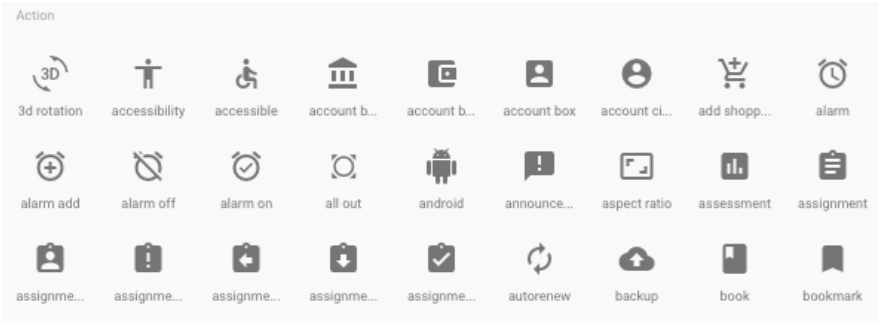


Figure 6: Google's material icons

3.3.4 Angular material

Angular Material is a UI component library for Angular JS developers. It helps to create consistent and functional web applications while maintaining the modern web design principles. Inspired by Google’s Material design, it helps creating fast and responsive websites and web applications.

Angular Material easily works with modules such as Flexbox and is relatively easy to manipulate. This helps with designing the application in the preferred style and lay out the contents without facing problems when making the application or site responsive. (Angular Material Tutorial n.d.)

3.4 Responsive web design

Responsive web design is the approach that suggests that design and development should respond to the user’s behaviour and environment, based on screen size, platform and orientation.(Smashing Magazine 2011) By applying responsive web design principles, it’s possible to make the product available to any user, no matter the hardware and software used, without losing the quality and functionality of the product. It is manipulated through

the use of JavaScript and CSS3 @media queries which allow to maintain the original design and automatically adjust it to the user's needs.

3.4.1 Screen resolution

Responsive web design is largely based on the available screen resolution, to which the design tries to adjust. The same design must be viewable and usable from variety of devices, no matter if the computer screen size is 11"in or 21"in or even if it's an iPhone 4 or a tablet that the website is being displayed on. However, it's impossible to make the design respond to every one of the thousands of currently available and used screen sizes. (Marcotte) Fortunately, a variety of sources online offer a list of the most common screen resolutions, allowing the process of adjusting faster. Additionally, browsers such as Google Chrome offer an in-built inspector tool that allows the designers and developers to test the different resolutions easily. (Marcotte 2010)

Top 10 Screen Resolutions		
1	640x360	19.22%
2	1366x768	15.07%
3	1024x768	8.32%
4	568x320	7.46%
5	667x375	6.70%
6	1920x1080	5.87%
7	1280x800	3.01%
8	1280x1024	2.77%
9	1600x900	2.68%
10	736x414	2.58%

Figure 7: Most common screen resolutions (W3Counter 2016)

By using a list such as in the Figure 7, and possibly the inspector tools, it's easy to make decision as to which screen resolutions the users are the most likely use the application and which are the best to be supported by the application before its' design is switched to the mobile design.

One of the approaches is, for example, the following. By using:

```
@media screen and (max-width: 1280px) and (min-width: 1024px)
```

it can be determined that a certain style will be displayed while the screen width is between 1024px and 1280px. Different min and max screen size queries can be specified, therefore allowing the style to be manipulated automatically as the screen resolution changes. However, it must be noted that overlapping values will display one of the possible styles, so it's important to avoid this by carefully choosing the values.

It's also important to think about the UX while creating responsiveness, and instead of just scaling the objects, it must be planned out accordingly if certain resolution requires a layout change. If scaled down to a mobile/tablet size, it’s good to consider creating a mobile version of the application right away instead of styling the web styles to fit the mobile.

3.5 Browser support

Although a lot of browsers are picking up the speed with supporting the most modern technologies, cross-browser compatibility is still a common issue, making the design implementation process a little more difficult. While it might seem that one code should apply to every browser, it doesn’t, and the same CSS can create slightly different results depending on whether Google Chrome or Opera browser is used for displaying it. (Techterms 2009)

Top 10 Web Browsers		
1	Chrome 51	38.35%
2	Safari 9	10.90%
3	Firefox 47	7.45%
4	IE 11	5.19%
5	Chrome 50	3.70%
6	Chrome 49	3.35%
7	Android 4	2.04%
8	IE 9	1.74%
9	Chrome 44	1.65%
10	Chrome 38	1.63%

Figure 8: Most commonly used browsers, including the versions (W3Counter 2016)

It's possible to use the most modern elements and styles in the design, but it should be kept in mind that a lot of users might not have the latest update of the browser, meaning that a lot of the newest technologies might be displayed broken or might not be displayed at all. To avoid that from happening, research must be conducted thoroughly for the newest elements

that will be used in the design. Once implemented, the final design should be tested on different browsers to fix or compromise as many issues as possible prior publishing.

If some values are different across different browsers, or if something doesn't respond using the same styling, special prefixes called Vendor Prefixes are available. These prefixes let the browser know which values should be read to ensure the correct display of the design. These prefixes are:

- -webkit- for Chrome, Safari and newer versions of Opera
- -mox- for Firefox
- -o- for older versions of Opera
- -ms- for Internet Explorer

(Vendor Prefix n.d.)

Example:

```
Button {
          border-radius: 2px;
          -moz-border-radius: 3px;
          -webkit-border-radius: 5px;
          }
```

This example shows how to get the appropriate button design on different browsers, using different values for the select browsers that may not display the design correctly. These prefixes can be added in cases when the browser might not recognise a certain declaration, as well.

3.6 Colour theory

When creating a design for a web application, the knowledge of colour theory can be crucial. By understanding the colours, it becomes easier to create visually appealing designs and the designer can make the users feel in certain way about the application.

In this section the general colour theory is explained as well as how it applies to web design, noting the similarities and differences. Additionally, it covers the psychology associated with the used colours in web design and how to select the the appropriate colour scheme.

3.6.1 Classic theory

Classic colour theory involves understanding how to best create harmonious colour combinations. This is important to avoid situations where the colours clash strongly, creating an unpleasant reaction in the users/viewers.

When learning about the general colour theory, there are several things that are always covered: colour schemes, colour meaning and the types of colour wheels. Though some are less important than others, all of these topics play a role in understanding how the colours work, making the designing process easier and the final result - better.

There are different colour schemes that can be used:

- Monochromatic colour scheme
- Analogous colour scheme
- Complementary colour scheme
- Split complementary colour scheme
- Triadic colour scheme
- Tetradic (double complementary) colour scheme

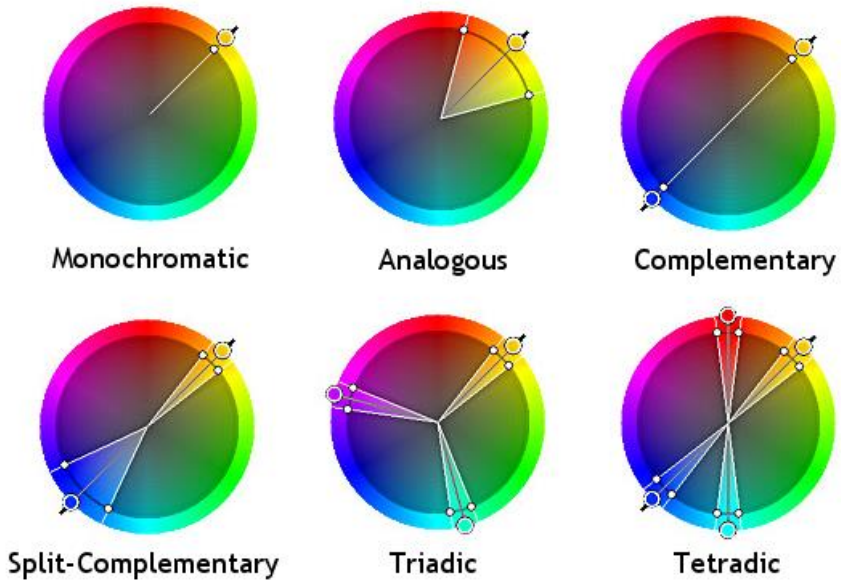


Figure 9: Colour scheme types (Color theory basics n.d.)

By looking at the Figure 9, it can already be understood what is the difference between these schemes. Monochromatic scheme focuses on the variations of a single colour using lightness

and saturation while Analogous uses also the adjacent colours to give more variety while maintaining the similar colouring amongst elements. Contrary, Complementary scheme uses the lightness and saturation variations of the opposing colours of the wheel. Complementary and Split-Complementary schemes tend to have more contrast, with a warm colour against a cold or vice versa, though Split-Complementary uses one colour and two adjacent colours of the complementary colour. Triadic colour scheme is only faintly different, with three colours from the wheel picked at equal distances from one another, but essentially it follows the same logic as Analogous, allowing more variations available while maintaining both - the slight contrast and the harmonious feeling. Tetradic scheme uses all four colours equal amount, with every two being complementary colours amongst themselves.

As aforementioned, another thing that is usually looked at as part of colour theory is the meanings of the colours. Although it might seem to many as something superficial, the knowledge of colour meaning can actually be helpful in design. Certain colours evoke certain emotions and it's a frequently used tool everywhere. In UI/UX design it's important to understand these, so the final product would gain the required reaction from the user. (Classic color theory n.d.)

Finally, there are two types of colour wheels: Mixing colour wheel and Visual colour wheel. Though generally the same, they serve different purposes. The Mixing wheel is more commonly used in arts, using the primary colour (Red, blue and yellow) to create a palette of mixed colours. The Visual colour wheel, meanwhile, is the more common representation of the way the colours are used in the modern technologies. The red, green and blue (RGB) colours or cyan, magenta and yellow (CMY) are the standards of the screens, cameras (RGB) and printers (CMY). (Visual vs. Mixing color wheel n.d.)

3.6.2 Web-oriented theory

When creating any designs for the web, it's good to understand that there are some guidelines that should be minded. The most common ones are the psychological aspect of colours the fact that there are parts of the classic colour theory that only partially apply, such as the usage of different colour schemes.

As mentioned already in the classic colour theory section, colours evoke emotions and it is important that the feeling that the design gives is the one intended by the website. What makes it a little more complicated it's that colours can mean different things in other cultures, requiring an appropriate research if the final design will be used for a certain audience. Bright colours can also strain eyes, and drive the visitor away. Similarly, if colours

used don’t represent the product in some way, it can cause a negativity or indifference in the visitor. (Cyriac 2009)

Simple and crisp colouring appeals more to serious audiences, giving the impression of cleanness and order. Meanwhile more colourful designs can appeal to younger audiences, as the brightness of the design promises fun and entertainment. So, for example, a children's website design, such as that displayed in Figure 10, should create joy and ease, meaning that light and bright ones should be used. Business-related and more serious websites, such as that in figure 11, meanwhile would go with a light UI and mostly muted colours, only using brighter ones for when something important must be accented. (Studio 1 design n.d.)

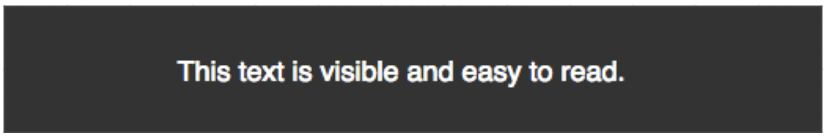
Figure 10: Childrens' websites use bright UI

Figure 11: Banking websites, such as Swedbank, use simple UI with some colour accents

Another important thing is that the colours must be combined careful and that not every colour combination works for the web. Such as in the following example:

Caption 2: Orange text is hard to read on a yellow background for a long time

As the Illustration 2 explains; although it's possible to read the text, it can cause a strain on if the visitor would have to read it for a longer time. This is a good example to prove that using analogous colours is not an appropriate approach when designing for web. Instead, it's better to focus on having a stark contrast between the elements, making the use of tints and shades extremely important.



Caption 3: White text on dark grey background is easy to read

A good example is white text on dark grey background or vice versa. The contrast is strong, but the colours are not too bright, making it extremely easy to continue reading. So this makes it easy to understand that also the colours used should not be overwhelming.

Caption 4: Red text on green background is strong contrast, but hard to read

Additionally, the contrasting colours must be picked with an additional care. Though dark red colour on a green background is a strong contrast, it creates a strain on the eyes when reading. The colours clash and the extremely strong saturation of both colours make it generally difficult to continuously comprehend the read text. (Cannon 2012)

4 Tools

In the design and implementation process, several different applications were used. This section lists and describes these applications, explaining why these were chosen for the project and the reasons why some of their use was discontinued.

4.1 Adobe Photoshop CC

Adobe Photoshop CC is a graphical editing software commonly used by designers and photographers. It’s part of the Creative Cloud service provided by Adobe Systems Inc. and stands alongside with such programs as Adobe Illustrator, which is a favored option for vector designs, and Dreamweaver, a program for web design and front-end development.

With its' many tools and user-friendly design, this tool was originally chosen for the project, tweaking the ideas for the designs. However, due to the budget limitations, this technology was used only briefly at the beginning of the project and other, open-source solutions were sought out to replace it.

4.2 Draw.io

Draw.io is an open-source Google Chrome software that can be used as an alternative to Microsoft Visio. It's used for making diagrams, flowcharts, process diagrams, org charts, UML, ER and network diagrams. The fact that it can be connected to the Google Drive makes it a comfortable tool to use, where the data can be accessed from any device with an active internet connection. With the various elements and tools offered, it can be used for simple designs of web applications as well. (Adobe n.d.)

Although not as well-suited for UI design as Photoshop CC, this was chosen as a fitting substitution for the wireframe and mockup creation.

4.3 GIMP

GIMP is a GNU image manipulation software. It can be used as an alternative tool to those such as Adobe Photoshop CC, even though the functions are limited in comparison to the Adobe product. It's available on all operating systems and supports a variety of file extensions. (GIMP n.d.)

This technology was deemed a suitable and cost-effective substitution for Photoshop CC. It was used in creation and alteration of the necessary design elements.

5 Languages and modules

In this section the languages and modules used in the project are described. Firstly, the CSS and Sass languages are covered, followed by a general description of the Flexbox module, explaining the advantages of applying these in the project.

5.1 CSS

CSS stands for Cascading Style Sheets and is a markup language used in HTML to style the elements how they are supposed to be displayed on the web UI. It works using tags in the .html file and describing the style rules in a <style> tag in the <head> tag at the top of the file or in a separate .css file that is linked to the .html file through the use of <link> tag.

Example:

Index.html

```
<head>
    <link rel="stylesheet" type="text/css"
    href="mystyle.css">
</head>
<body>
    <h1>Title of the page</h1>
</body>
```

mystyle.css

```
body {height: 100%;}
h1 {font-size: 15pt;}
```

(W3Schools n.d.)

CSS is the only styling language for the HTML that doesn't involve the use of scripting languages such as JavaScript, therefore it was the chosen technology for the project. It has numerous extension languages, such as Sass and Less, and the standards are gradually updated and improved with versions, the current being CSS3, which is backwards compatible, making CSS generally flexible.

5.2 Sass

Sass is CSS extension language created by Hampton Caitlin and currently maintained by Nathan Weizenbaum and Chris Eppstein. It's essentially an alternative way of writing CSS and its' files are defined by the .scss/.sass extension. After saving, all the .scss content is automatically translated to a single .css file. It follows the DRY principle, which stands for ''Don't Repeat Yourself'' and, as the name suggests, tries to avoid unnecessary declaration duplication for different styled elements. (Catlin & Catlin 2011)

The advantages of using Sass preprocessor is that it allows the use of variables, nesting, extending and importing among others. Using variables makes the standardising of the styles for the implemented design easier, such as the following:

```
$primary-color: #FFFF00;
```

```
.yellow-bg {
background-color: $primary-color
                    }
```

By defining the colour #FFFF00; as $primary-color, it can be used in all the associated .scss files using that name. The use of variables doesn't stop at colours and can be used in different declarations, such as font-size, padding and many more. This, as mentioned earlier, helps with maintaining the similarity in the elements of the design as well as simplifying the styling process.

Another one of the advantages of using Sass is nesting. It allows for the general comfort and visibility that the regular CSS doesn't offer. Instead of writing every styleable element separately, they can be connected through the use of sub-declaration blocks, grouping the connected elements effectively. So, instead of declaring the style using separate blocks:

```
.body { height: 100% }

p { padding: 10px; }
```

It can be nested together:

```
.body {
Height: 100%
            & p {
               padding: 10px;
               }
            }
```

As the amount of declaration blocks increases, this method of writing becomes a valuable asset. Nesting can be useful also in the case of a certain element being used on several pages or in even in the same page, but requiring partially or completely different styling. It makes it easier to style that specific element without overriding the style of the similar element on another page or in a different context on the same page.

```
.left-article {
            p {
            color: black;
            font-size: 24pt;
            }
}

.right-article {
            p {
            color: red;
```

```
            font-size: 12pt;
            }
}
```

Nesting comes in hand with extending when applying the DRY principle. Instead of rewriting the same declarations in several objects, they can be extended from one object to the other, making the .css translate it accordingly.

Example:

```
.test1 {
        color: red;
        border-bottom: 1px solid green;
        }

.test2 {
        @extend .test1;
        background-color: blue;
}
```

Is translated in css as:

```
.test1, .test2 {
        color: red;
        border-bottom: 1px solid green;
        }

.test2 {
        background-color: blue;
        }
```

Importing is another important advantage of Sass. Although importing is a command that exists in CSS, allowing the CSS to be split into several smaller files, it requires for each individual import to send a HTTP request. Meanwhile Sass loads the file together with the file it's being imported into. This makes the loading speed considerably quicker, not hindering the comfort of writing the style for the sake of performance speed.

Sass has more advantages, but the ones described in this section were the main ones that made it the preferred technology in this project.

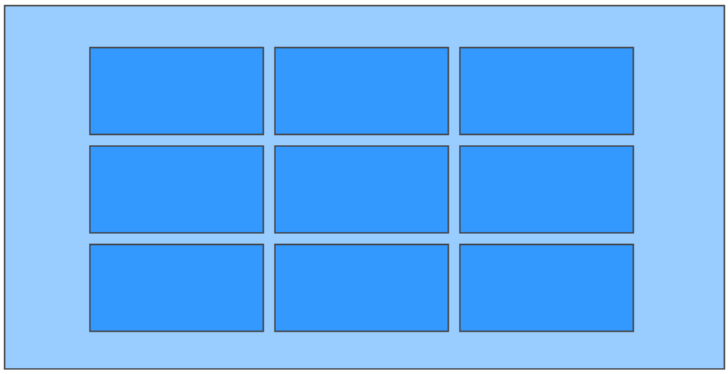
5.3 Flexbox layout

Flexbox Layout or CSS Flexible Box Layout Module, but commonly known simply as Flexbox, is a module for more efficient layout and alignment distribution in a container even when the item size is unknown. (Stojanov 2015) The advantage of using Flexbox is that it allows for more complex applications and webpages to be made. By having a flexible layout, it’s possible to align items and use the given space more efficiently. (W3 2016)

The way Flexbox works is based on the use of parent and child elements, also called containers, where every parent element contains one or more child elements that can be manipulated in several ways:

- They can be laid out in any flow direction,
- They can be displayed in reversed or rearranged order at the style layer,
- They can be laid out or wrapped into multiple lines along a single axis,
- They can be ‘flexed’, to respond to the available space,
- They can be aligned with respect to their container or each other,
- They can be dynamically collapsed or un-collapsed along the main axis without leaving the container’s size.

(W3 2016)



Caption 5: Parent-child interaction in Flexbox

Displayed in the Illustration 5 is an example of how the parent-child interaction works, and how it is made easier to manipulate when using Flexbox. The 9 child elements (Dark blue) are aligned in the perfect center of the parent element (Light blue), with the row snapping according to the restrictions of the parent’s set width and height and not letting the child elements to go outside the parent element or snap partially into the next row. Depending on the declarations in use, these elements could be displayed in a horizontal (row) or vertical (column) direction and as long as the parent has strict size declarations, the elements will be displayed correctly. The problem may arise if both, the child and the parent have strict size declarations and the amount of child elements exceeds the container possibilities of the parent. This, however, can be avoided by planning the layout in advance. These size requirements are important to be able to use other flexbox declarations, as the module must know the available space before it can align and justify the elements according to the needs of the layout.

The flexibility of this module is suitable for implementing the design on a web application, as the Flexbox rules are less likely to allow the items to scatter across the page when faced with

responsiveness. This also enables generally more complicated designs to be implemented more efficiently.

6 Practical implementation

This section describes the practical implementation of the previously-described technologies and knowledge base to create the wireframes and mockup of the design. Additionally, it covers user testing, in the form of a feedback survey, the analysis of these results and the implementation of the changes in the design based on the analysis results.

6.1 Benchmarking

Benchmarking was included in the implementation process and was done in collaboration with the customer. Several different dashboard UIs were reviewed, mapping out the details and practices used and applying them to the design of Lumi platform. This section covers the both benchmarking phases in the project, explaining the originally used elements and which were changed or removed after the first pilot test.

Most of the final decisions were based upon three main dashboard designs that were observed, though approximately 10 were originally listed as possible references/guides for the platform's design.

Figure 12: Analytics dashboard

The Dashboard UI design by Dmitry Kornushin on behance was one of the biggest inspirations used in the creation of the original design. The general layout, the navigation style and the

tabbing between the different time frames was inspired by those seen in the Figure 12 Originally the user information such as the username were also included in the header of the application, based on the benchmarking results, only to be removed onwards in the design process, even before the first test was conducted, instead set aside as a separate section in the application.

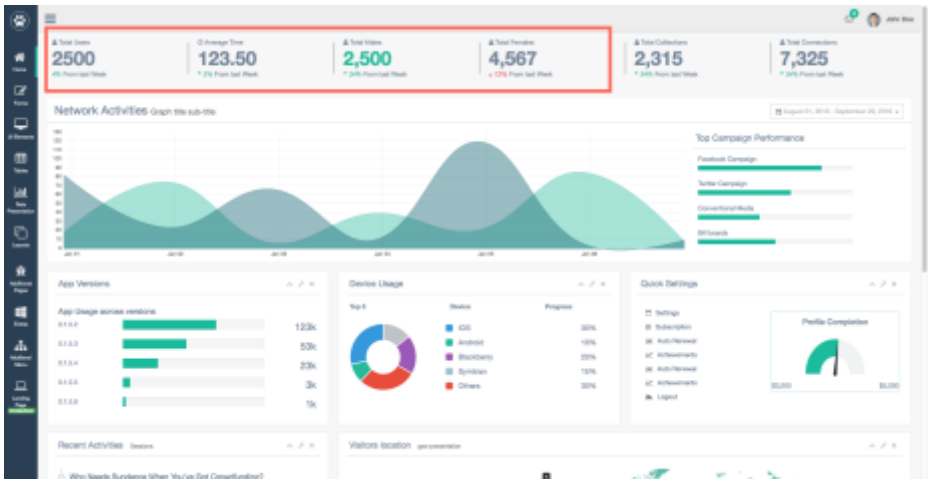


Figure 13: Gentallela Alela! dashboard

A common detail in many of the dashboard examples that were viewed during the benchmarking process was the use of separate section at the top with either the total numbers or tendencies, such as that shown in Figures 12 - 14. This detail was incorporated in the original design, but was removed onwards as the benefit was limited and the same information could be received in the report section later on added in the application.

Figure 14: Lumino admin board

When benchmarking for the design improvements post-test, several unnecessary elements were removed from the dashboard of the application. Instead the dashboard was made clearer, ensuring that only actual essential information is displayed in it. Similarly to Lumino design in Figure 14 the main graph was set to take over majority of the first view in the application, with minor graphs displaying the totals and tendencies alongside with navigation calendar. In addition with the navigation style taken from Gentallela Alela!, the goal became to make the information as easy to view as possible.

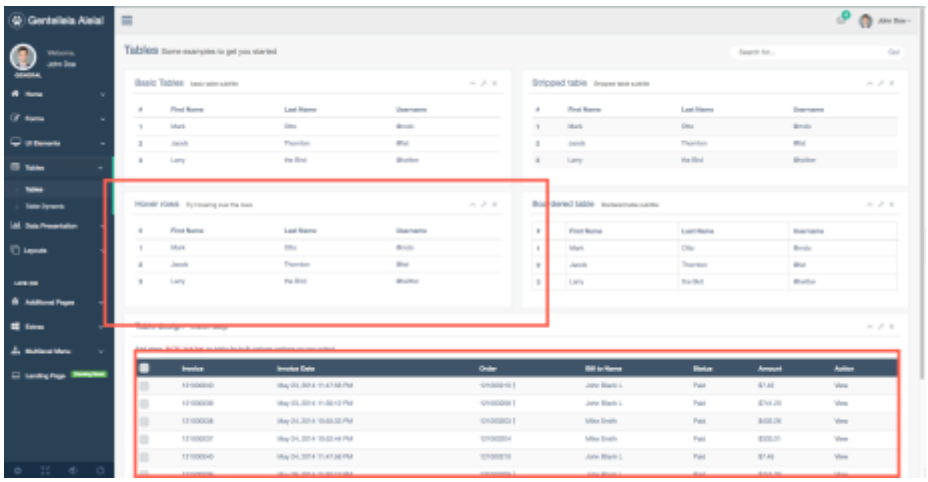


Figure 15: Table styles

Though some other elements, such as the table styles seen in Figure 15, were based on the benchmarking of precisely those elements, largely the rest of the designs were created based on the ones established in the dashboard page of the application.

Some final mutual requirements were achieved in the benchmarking process, differing between the two versions of the application design. The first mutual requirements were the following:

- The dashboard displays:
    - Graphs of the most relevant information,
    - Total values,
    - Real time logs,
    - Calendar for navigation,
- The day/week/month values are navigated through tab selection,
- Side navigation is created in the manner seen in Analytics dashboard:
    - Buttons use text and by-standing icon for description,
    - Subtle colour change,
- The colour scheme would mainly be light with a dark navigation.

After the first pilot test of the application and the second benchmarking, the following mutual requirements were:

- Large, clear graph is used in the dashboard,
- Side navigation buttons use larger icon with smaller text beneath for description,
- Subtle, but more noticeable side navigation colour changes,
- Light general UI with only a dark side navigation and buttons,
- A separate drop-up with selection available on dashboard and control panel pages,
- White container for every separate element,
- All the secondary information moved to other pages:
    - Total values moved to reporter,
    - Logs moved to control panel drop-up.

6.2 Wireframe

When creating the wireframe, the first thing that had to be done was listing all the intended functions for the web application as well as figuring out what should be included in the main focus and what should remain secondary. Some of the elements that were decided upon firstly for the reporter were:

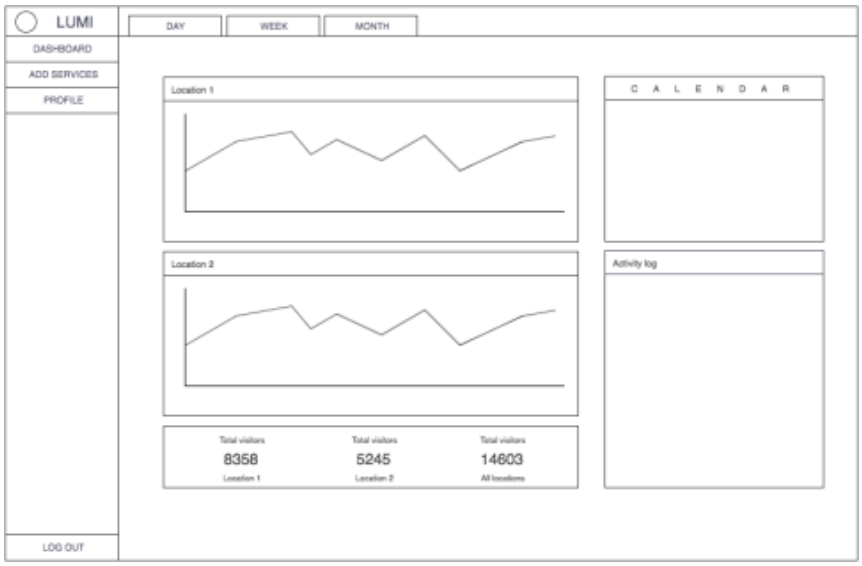
- A login page with ''forgot password'' functionality,
- Different people counting data charts based on time,
- Live log of the incoming data from the locations,
- Total amount counters for locations,
- A profile page with the option to change password and edit contact information,
- Options to add new groups, devices and users.

Using this list, it could be deduced which functions should be primary in the application and how they should be displayed. Due to the focus on gathering data, several pages were decided upon and the order in which these would be displayed, as well.

Page 1: The login page, including the button leading to the adjoined ''forgot password'' page. This page is the first one any user sees and it's not possible to bypass it to see the application. This implies that data security is a number one priority and the clear, simple login design allows the user to understand the function without the need to think long.

Caption 6: Login page wireframe

Page 2: The dashboard page, displaying the main information that the user would be interested in seeing – the data charts, live log and the total people amounts for the locations connected to the device adjoined to the account. A calendar for easier navigation also was included during the wireframing process. The application can be navigated through the side navigation bar as well as using the tabs in the header that allow viewing different time frames of day, week and month basis.

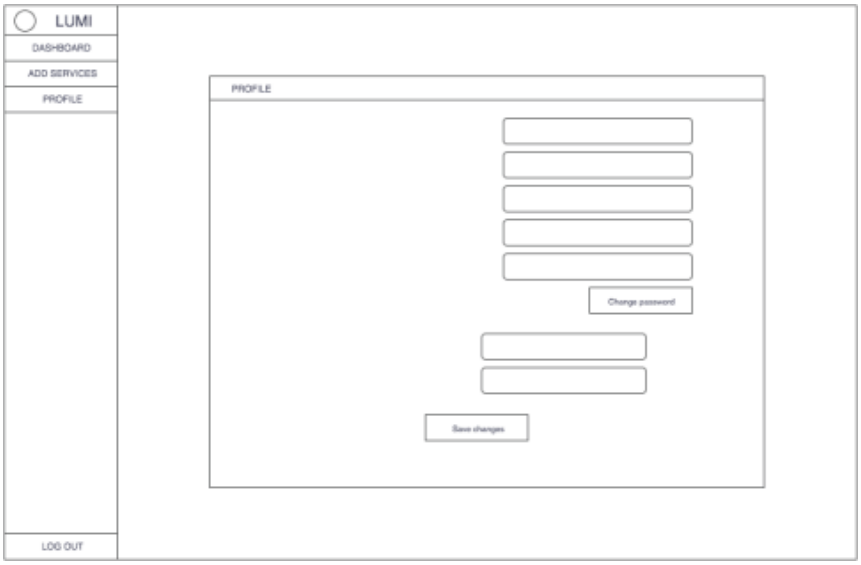


Caption 7: Dashboard page wireframe

Page 3: The adding of new users, devices and groups happens through a popup window allowing the user to select what would they like to add by navigating through the use of tabs.

Caption 8: "Add" popup field wireframe

Page 4: The profile page displays the account information that cannot be changed, such as the login ID, and the information that can be changed, such as the name and the contact information. Additionally it has an option to change the account password.



Caption 9: Profile page wireframe

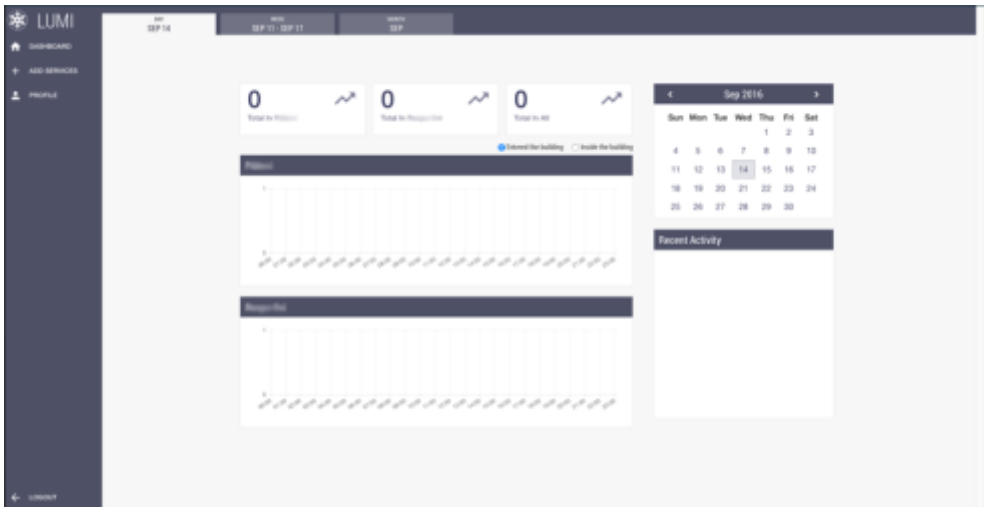
It was decided that the options to add users, groups, devices and the bug report will presented in the form of a popup fields. These would be separate between the add options and the bug report, where add options would be all in a single popup with tabs as the navigation between the three add options. The bug report was decided to be a separate window with the options of either submitting a bug report or giving the user the option to submit some possible requests for the application.

6.3 Mockup

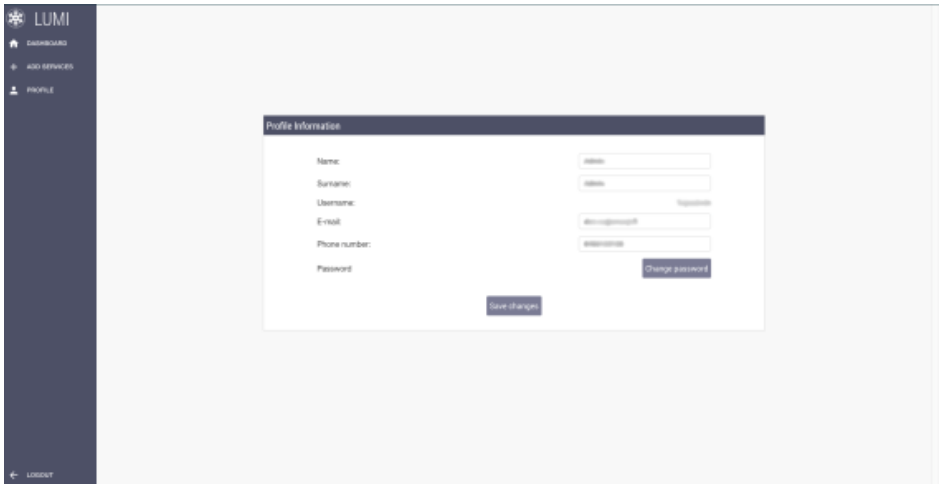
In this section the process of the mockup creation is described. Firstly, it covers the decisions made about the style in which the application would be thereafter created. Afterwards it goes to explain the colour palette selection and changes, the used icons and fonts, as well as the approach for designing the functional application elements.

The final result described in this section are the following mockups:

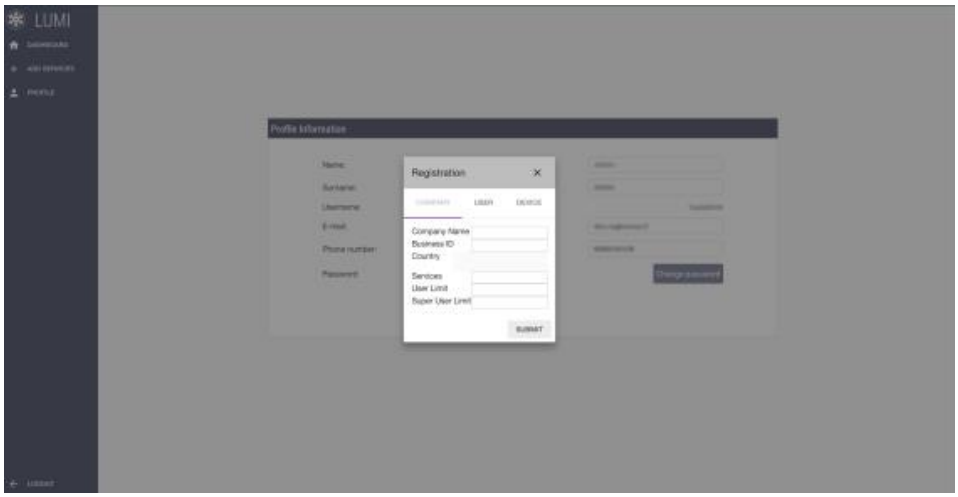
Caption 10: Login page



Caption 11: Dashboard page



Caption 12: Profile page



Caption 13: "Add" popup

6.3.1 Web application style

When creating the actual mockup of the application, the first thing that had to be decided upon was which style would be used. After some research and discussion with the company, the decision was soon reached, selecting Flat design as the main approach in the application. Using the wireframes created earlier in the designing process, the mockups were created.

6.3.2 Colour palette

Firstly, the colour palette was based on the originally intended company brand colours with purple being the main one and white – secondary. To add variety and contrast, two types of dark and one light colour were added for the use with the text and other elements that would require them.

Caption 14: Original colour palette

The colours were soon brought under the question again when the first mockups were created, as the original palette didn't fulfill the requirement of maintaining the professionality and took the attention away from the important information in the application. Therefore the use of the purple was left to minimum, only remaining in the design as one of the colours used in the charts and as accent in the material design based elements. The grey tones were changed during the process, too, making them lighter and more neutral, but maintaining the necessary contrast by maintaining the blue tint lost when minimising the use of the purple. By using opacity with the white colour, a certain colour change could also be achieved.

Caption 15: First design colour palette

Using the changed colour palette, the application mockups were created. This colour palette is also the one that was used in the pilot prototype of the application.

6.3.3 Icons and font

In the creation process, several options for the icons were discussed, the final decision being that Google's material icons would be used in the project. This decision was based on the fact that these icons could easily be integrated into the project from the technical side, and due to the offering fitting the needs of the design. Though the icons are generally similar, it still required careful selection process to choose which icons to use for each section and whether to pick the full or the hollow ones.

Due to the Flat design's focus on making things crisp and easy to read, appropriate fonts had to be selected. Google fonts were searched to find some sample fonts to use for this project as it's necessary that the font is not only fitting for the design, but is also well-supported across the browsers and offers different size options and transformations. One of such fonts is Open Sans; a widely used font and previously already utilised in one of the projects in the company. However, it was soon decided that Roboto and Roboto Condensed would be the

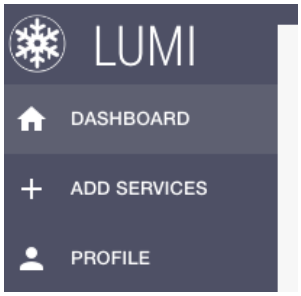
implemented fonts in this project due to the wide usability, support and the similarity between the font families, allowing two different fonts to be implemented without having too much, unnecessary differences.

Caption 16: Roboto and Roboto Condensed fonts with sample comparison

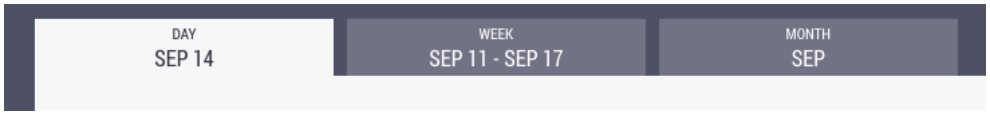
6.3.4 Mockup functional elements

One of the most difficult things in the design process is making the functional elements appear so while maintaining the flat design principles. Since flat design doesn't include the use of shadows or other 3D qualities, buttons were deemed to show their functionality through a colour change upon hovering over them with the pointer, otherwise implying their functionality through the contrast that makes the element stand out. Tabbed elements change the colour or have a colour indicator according to whether or not they are active.

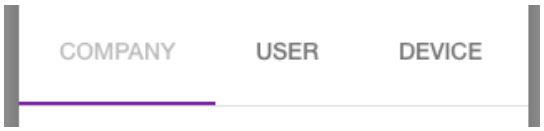
Caption 17: Buttons change colour when hovered over



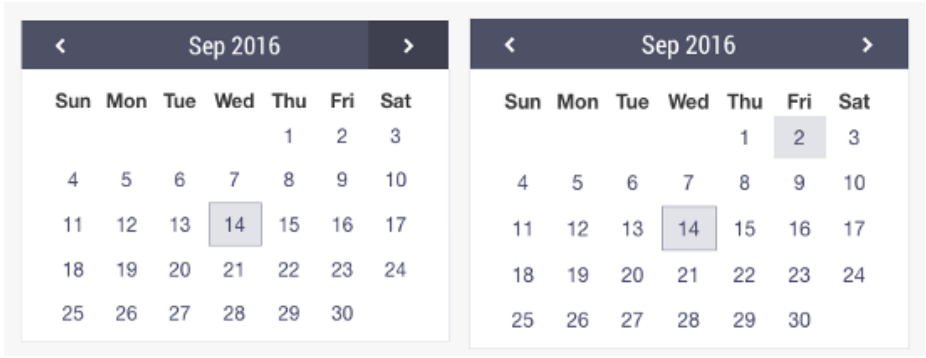
Caption 18: Navigation buttons change colour when hovered over



Caption 19: Tabs change colour upon being activated



Caption 20: Popup active tab colour indicator



Caption 21: Calendar buttons change colour when hovered over

6.3.5 Responsiveness

Making the application responsive was one of the main requirements in the project when it came to design implementation. It was decided that it would scale down to be responsive to a certain tablet resolution and anything beyond that would remain second plan for the future of the project. As such, the resolution widths that the application was made to respond to were the following:

- Min 765px – max 1000px,
- Min 1000px – max 1280px,
- Min 1280px – max 1365px,
- Min 1365px – max 1590px.

As it is, most elements in the select resolutions could be scaled and rearranged flexibly. Only some of the elements were affected otherwise, either disappearing or visibly changing the size and content.

Firstly, at the lowest resolution the calendar element from the dashboard had to be removed. Instead, in future, this element would appear as a button, allowing the user to select the date from a popup calendar. This decision was made due to the, at the time inflexibility of

the calendar and due to the space restrictions on the page, eliminating the displayed elements in the manner of least importance at the time.

Secondly, the lower resolutions also included the scaling the side navigation width and therefore – the buttons. Instead of displaying the buttons with the appropriate text next each, only icons were displayed. This proved a challenge for the design, requiring that the icons used would be as descriptive as possible.

As final difference is the disappearance of the icons used in the ‘’Total amount’’ boxes, allowing the space to be used more efficiently and avoiding unnecessary scaling of the important information displayed. Though technically the space allowed for the use the icon, it was deemed unnecessary as the original function of the icons was decorative.

6.4 User testing

As part of the web application creation process, a user testing was conducted at the end of the first real-life pilot of the application. The pilot test was conducted for about three months using the design displayed in Appendix 2: First mockups. After the three months of the pilot test, a 1UP Media representative met with the test environment representative and conducted an interview using the Discussion Guide prepared in advance. These answers where further on used to improve the web application design and functionality further.

By the end of the pilot test, additional functionality also was developed and was required by the company to include in the updated version of the application design.

Due to the pilot testing environment company being fully Finnish, the help of the 1UP Media Oy staff was involved, translating the questions to Finnish and the feedback given by the test users - back to English.

6.4.1 Analysis of the feedback

While analysing the feedback, it was clear that there were parts that the test user considered to be good and some that could use some improvement. Though there were some functionality issues that occurred during the test, the user feedback was mainly positive and constructive, allowing for the improvements to be implemented in a much easier manner.

As per the questions seen in the Appendix 3 the following things were considered at the beginning of the interview:

- The ease of understanding the layout and navigation,
- The ease of understanding the functionality,
- The response to the colour scheme,
- Any issues with the design and its functional elements,
- The response to the offered elements and functionality,
- Ideas for any additional functionality and changes in design.

These questions were used to best give the understanding to the test user of what sort of information would be the most valuable for improving the product, but also giving enough of open-ended questions in case of additional ideas that the test user might have. Using the answers, it could be determined that the chosen colour scheme was considered appropriate for the application and that, generally, the application was clear and easy to use. However, not considering the general issues discovered with some functionality of the application, some improvements were definitely necessary such as including a wider variety of responsiveness as there had been problems with the application when it was viewed on a smaller laptop screen. An issue that was soon discovered included the functionality of buttons, the response of clicking restricted only to the text within the button.

6.4.2 Required improvements

After the analysis of the feedback and the changes decided upon by the 1UP Media Oy, the following improvements were required and were agreed upon to be implemented in the web application:

- Changing the dashboard layout and including the location-based chart,
- Adding a drop-up function with easier selection of the preferable viewable devices,
- Move the real-time log to the drop-up,
- Add a user list associated with each device,
- Add the option to create reports and to export them in different file types, including PDF,
- Create a user feedback page,
- Create a help section with Frequently Asked Questions,
- Change how the add function works,
- Make slight changes in the colour palette,
- Make the design responsive for different screen sizes.

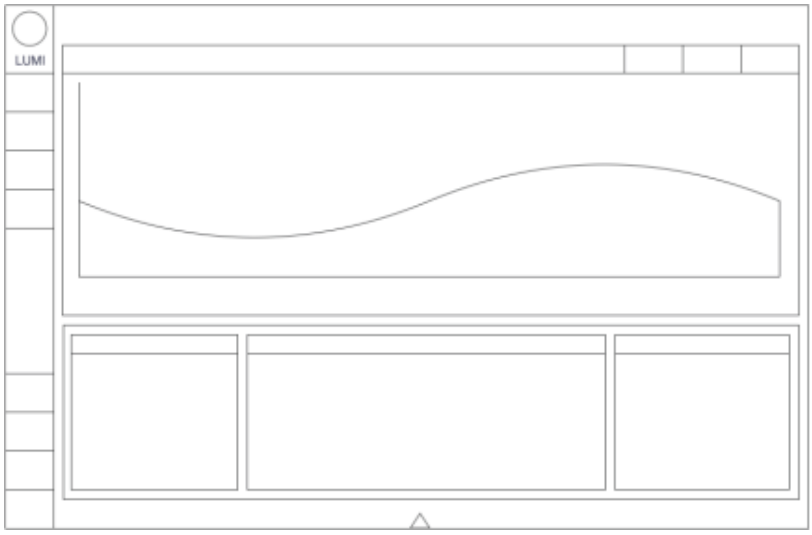
With these changes, additional wireframes had to be created to plan the new layout of the already existing pages and the new ones. The design of the PDF also had to be planned, due to the restricted nature of the file type.

6.5 Final design

To change and improve the newest design, all the application had to be brought back to the wireframing phase, though the core functions were not changed. This section will look at the planning of the new additions to the application, the redesigned wireframes and the final implementation of these designs into the application.

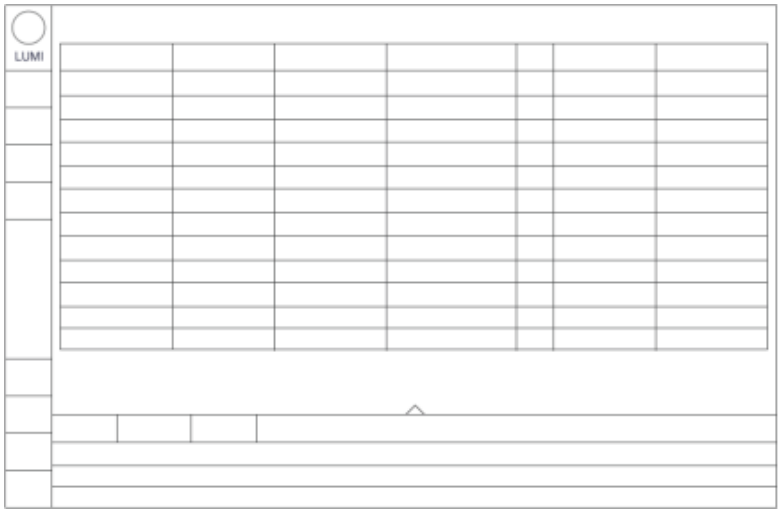
6.5.1 Wireframes

This section covers the process of creating the new wireframes for the improved application based on the user testing and customer feedback and requirements. It goes through describing the Dashboard, Control panel, Report, Profile, Feedback, FAQ and Bug report pages. General changes that affect all the pages in the application are described separately. The full wireframes can be viewed in Appendix 4: Improved wireframes.



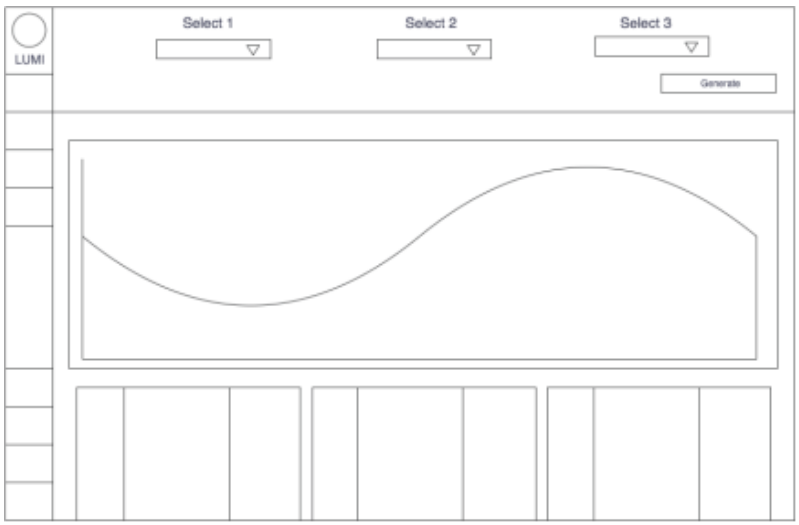
Caption 22: Dashboard page wireframe

As one of the first changes made in the dashboard page, the live log was removed and the time-based charts were merged into one large chart with better tab options for navigation between time frames of day, week and month. A location-based chart and a year total chart were added at the bottom section of the page. A dropup function at the very bottom of the page allows the user to select which location data to display on the dashboard. Additionally, a list of live logs and details about the location are available in tabs.



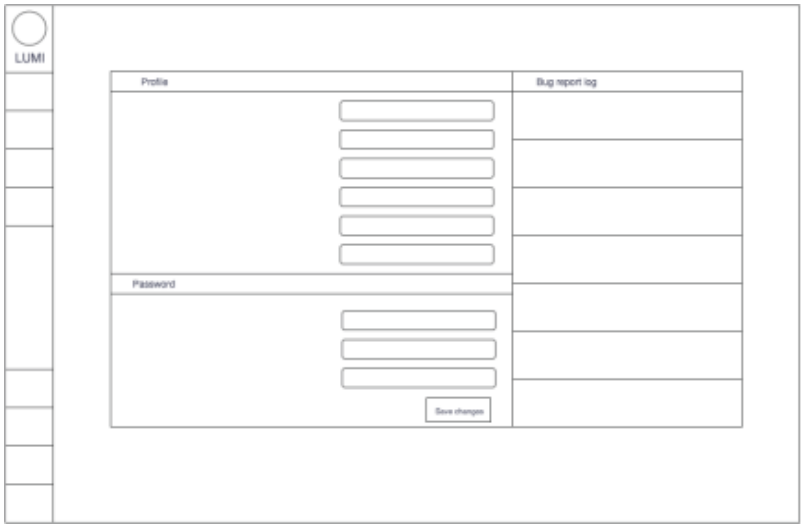
Caption 23: Control panel wireframe

An addition to the application was a separate control panel page, listing all of the locations, devices and users, which can be searched using dropdowns and search fields and navigated through the list to either view more details about locations of a device or to navigate to the dashboard layout of all or select data recorded in the locations. Information of locations and devices can be modified and the devices can be manually enabled/disabled through the dropup.



Caption 24: Report page wireframe

Report page was added as per suggestion of the test users. The page is separated in two sections, the top allowing the user to select different parameters by which to query the report whilst the bottom part displays the generated results in a form of graph and tables for raw data. Additionally, buttons for different download/print options are added so the user can download either a .csv or a .pdf file with the information generated.



Caption 25: Profile page wireframe

Profile page had minor changes and one major addition. The layout was slightly rearranged to make the process of viewing and editing the profile information clearer. The major addition is the bug report log, where the user can view the submitted bug report and request status.

For easier feedback gathering a feedback page was added. The user receives a basic feedback form with a list of questions prepared by the parent company of the application (1UP Media Oy), responds and submits the answers for future improvement of the application.

A FAQ page was included in the new design. In this page the information is separated in sections, each of which cover a certain topic, different questions listed underneath the topic title. One question at a time may open, automatically closing when the next is opened.

Bug reports may be submitted through the popup window similar to that of adding the new services. These submissions later will be displayed in the user profile page under the bug report log section.

The general navigation was changed as well, to ensure that it's easier to use and even more clear than the one used in the pilot version. It was decided that it will be shrunk in width, making it more compact, the button functionality not limited to only the text part of the button. The navigation is separated in two parts: the main functionality at the top and the secondary options – at the bottom. Additionally, the adding of users, groups and devices were separated, a main button of ''Add'' revealing the three sub-buttons to pick one of the possible options before opening it in a popup for filling in the required details.

6.5.2 Colour palette

Though considered good enough by the test user, the colour scheme was changed according to the new design, completely removing the originally user purple and expanding it more on the monochrome scale. The new colours picked ensured better contrast and subtlety of the overall scheme allowing the required information to stand out better.

Caption 26: New colour palette

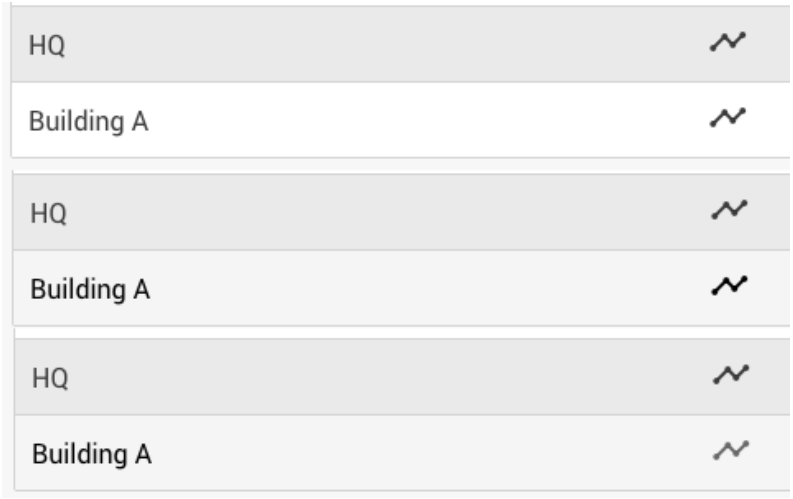
To help with the informationa being displayed in a manner that draws the user's attention to itself, an appropriate graph colour scheme had to selected. Though the colours selected are not following the monochrome theme mainly maintained in the design, the saturation allows the graphs to stay noticeable but not obnoxious.

Caption 27: Graph colours

6.5.3 Functional elements

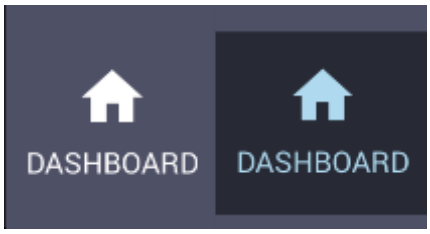
Though the elements had an approach for how to display that they were functional, the changes and additions in the overall application required some changes in how to display it. Though the change of colour according to the activity status and hovering is still in use, some elements had dual functionality.

One of such elements are the list elements in the control panel where the clicking on the text and on the graph icon caused different effects. At the same time, the control panel elements could be generally however above to make following the viewed list easier. With this, there was a necessity to ensure that the user could understand that the location name and the graph icon could be clicked , which was achieved by the colour changing of the text/icon upon hovering over it with the cursor. Though subtle, in fast-paced environment, the changes are easy enough to see and understand.



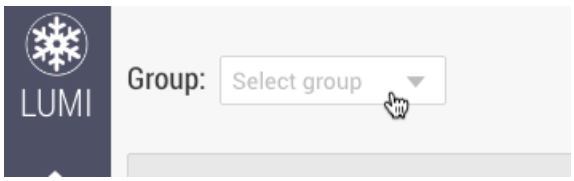
Caption 28: The changes of the functional elements in the table

Similar approach was taken with the navigation buttons which previously only reacted when hovered over. The main purpose of navigation is to help the user navigate the application and it should make it clear at all times where in the application the user is. With this, another change that was implemented into the new design was that the active navigation changes the colour to indicate the location better.



Caption 29: The active navigation button colour change

As the final addition that was made to help the functional elements to be more visible, the cursor was declared to show as pointer when hovering over functional elements, such as dropdowns. It's a common practice on many websites, even where other qualities of the element display their difference from static elements, and it's widely recognised by the general audience, making it the most useful styling tool in this case.



Caption 30: Cursor to pointer change

As displayed in the Illustration 30, another, very clear way to show that an element could be active in case of dropdowns and input fields is by having a placeholder text that tells the user to do something. Though it might not bring the attention of the user to it, it definitely speaks of a function hidden within it.

6.5.4 Responsiveness

One of the issues mentioned by the test user was the fact that the application did not have an appropriate responsiveness, making it difficult to use it on the test user's laptop. This meant that more attention had to be given to making the new design respond to more potential screen sizes. Several width specifications were decided upon for the future tests, most of the sizes remaining the same, with small changes in the available ones and splitting one of the previously used size range into two:

- Min 700px – max 1000px,
- Min 1000px – max 1024px,
- Min 1024px – max 1280px,
- Min 1280px – max 1365px,
- Min 1356px - max 1590px.

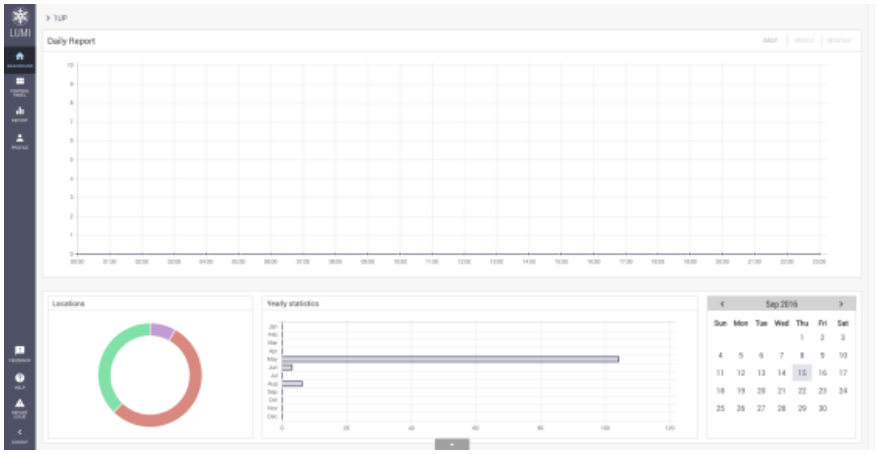
The new elements used were a lot more flexible, their layout in the application a lot easier to manipulate. This allowed the changes to respond a lot more actively and made it easier to avoid the need to compromise by removing elements.

One of the remaining similarities between the old design and the new one was the removal of the calendar element as it is displayed originally in the design. It is, still, to be replaced in the future with an element that can be opened and hidden upon the user's choice, allowing to maintain the quality and user friendliness of the design. This was due to allow the important information to be displayed without the necessity to scale the graphs too small.

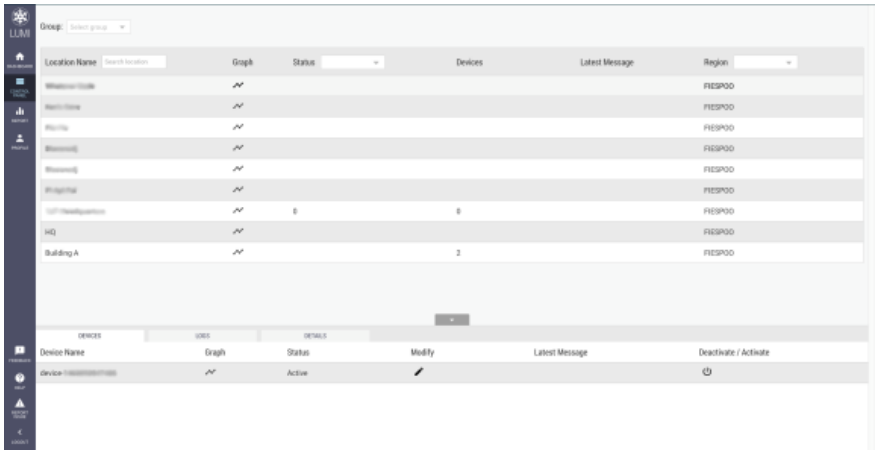
Most other elements in the application were, as aforementioned, a lot more flexible, automatically scaling according to the resolution, without losing their quality and user friendliness. Certain fonts had to be scaled down in size for the lowest resolution, as well as some of the margins. At the current project phase, additional responsiveness is planned to be implemented at the earliest convenience.

6.5.5 Final implementation

The final implementation process was carried out from August 2016 till 15th of September 2016 and the full final implemented designs can be viewed in Appendix 5: Improved mockups. Due to the thorough plan and careful use of variables and previous elements, the implementation of the new design was fairly quick. All of the changes were approved by the customer company and are to be tested and evaluated in the future.



Caption 31: The new dashboard design



Caption 32: Control panel with the dropup displaying devices

Caption 33: The report generator

7 Evaluation

Due to the fact that the Lumi application is still in the implementing stage, to prepare it for the second pilot test and to finalise the documentation associated with it, no data about the influence of the product on the business is yet available for evaluation. The main objectives were fulfilled and the appropriate design, based on the customer requirements was created.

Firstly, using the literature research, it was determined what made for a modern web application and this knowledge was applied in practice. The application is designed following mainly flat design principles. It follows a unified style across all the pages and uses the practices observed in the benchmarking process, to raise familiarity and ease of use for the user, which was onwards improved using user testing. The colour scheme was choices appropriately so it wouldn't create the strain for the user's eyes and would still evoke the clear, business-like impression of the application.

During the project, using the UCD framework, the design creation went through all the appropriate stages: planning, designing (Which includes the following: wireframing and mockup creation), evaluation (Both, during the design phase and the pilot testing) and implementation of the approved designs into the prototype. This allowed to reach the most optimal result for the end product at the time of the project end.

As the final of the objectives that was reached, the design was made responsive using the most popular screen resolutions. Unfortunately, the full compatibility for the browsers were not reached and had to continue beyond the project schedule.

8 Conclusion

Creating a modern and responsive design for a web application is an essential part in our increasingly computerised world. By following trends and understanding the basic psychology and theory, it's possible to create and improve designs as they are needed. Additionally, by conducting user testing, it's possible to understand how the user interacts with the application, allowing for further improvements.

The project was a success and the design created during this project will prove useful as the company's product progresses. Using the established designs, the company should be able to establish their role in the field and advance as further changes might be required. By understanding the process originally used in the creation of the design, the essential parts of what makes the brand style can be maintained and any rebranding can be easily applied to the existing design.

The company can utilise the designs in the future as it advances in expanding their cloud service. The designs for the people counting module were made to be reusable and easy to modify, should they chose to continue using the designs.

Whilst satisfied with the final design, I feel that it could be beneficial to explore lighter UI in the future. Though the current design is good for the business needs, it might seem heavy to the users so other possibilities and incorporating more colours might benefit the product in the future. Additionally, it must be ensured that the design remains unified as it may sometimes be difficult to use the same approach for different functionalities. Lumi platform is a new software which will have additional modules added to it as the time goes, making the design process more complex, but I am positive that it will be possible to maintain a clear image of it being the same roduct.

References

Angular Material Tutorial. No date. Tutorials Point. Accessed 24 September 2016. https://www.tutorialspoint.com/angular_material/

Browser & Platform Market Share. 2016. W3Counter. Accessed 28 September 2016. https://www.w3counter.com/globalstats.php

Cannon, T. 2012. An Introduction to Color Theory for Web Designers. TutsPlus, 12 September. Accessed 27 September 2016. https://webdesign.tutsplus.com/articles/an-introduction-to-color-theory-for-web-designers--webdesign-1437

Catlin, M. & Catlin, H. 2011. Pragmatic Guide to Sass. The Pragmatic Bookshelf.

Catlin, H., Weizenbaum, N., et.al. No date. Sass basics. Sass-lang. Accessed 24 August 2016. http://sass-lang.com/guide

Classic color schemes. No date. Color-wheel-pro. Accessed 24 August 2016. http://www.color-wheel-pro.com/color-schemes.html

Color theory basics. No date. Color-wheel-pro. Accessed 24 August 2016. http://www.color-wheel-pro.com/color-theory-basics.html

Cousins, C. 2015. The Evolution of Flat Design: Muted Colors. Design Shack, 18 November. Accessed 20 August 2016. https://designshack.net/articles/graphics/the-evolution-of-flat-design-muted-colors/

Cross browser. 2009. Techterms, 2 January. Accessed 27 September 2016. http://techterms.com/definition/crossbrowser

CSS Flexible Box Layout Module Level 1. 2016. W3, 26 May. Accessed 6 June 2016. https://www.w3.org/TR/css-flexbox/#intro

CSS Introduction. No date. W3Schools. Accessed 10 August 2016. http://www.w3schools.com/css/css_intro.asp

Cyriac. 2009. Importance Of Color Selection In Web Design. TechWyse, 23 February. Accessed 28 September. https://www.techwyse.com/blog/website-design/importance-of-color-selection-in-website-design/

Esparza, E. 2013. Understanding Website Benchmarking: When Is It Useful, When Is It Not? Market8, 22 August. Accessed November 1 2016. https://www.market8.net/b2b-web-design-and-inbound-marketing-blog/understanding-website-benchmarking-when-is-it-useful-when-is-it-not

FlatUIcolors. No date. Accessed 15 August 2016. http://flatuicolors.com

Gimp. No date. Gimp. Accessed 10 August 2016. https://www.gimp.org/

Greever, T. 2013. Flat Design is Not the Answer to Skeuomorphism. TomGreever.com, 16 August. Accessed 15 August 2016. http://tomgreever.com/flat-design-is-not-the-answer-to-skeuomorphism/

Mandelbaum, M. 2014. Design Principles. Learn Design Principles. Accessed 10 August 2016. http://learndesignprinciples.com/

Marcotte, E. 2010. Responsive Web Design. Alistapart, 25 May. Accessed 27 September 2016. http://alistapart.com/article/responsive-web-design

Material design. 2014. Google. Accessed 2 August 2016. https://material.google.com/#

Mock-up. No date. Merriam-Webster. Accessed 25 August 2016. http://www.merriam-webster.com/dictionary/mock%E2%80%93up

Notes on User Centred Design Process (UCD). 2008. W3C. Accessed 11 October 2016. https://www.w3.org/WAI/redesign/ucd

One-on-One Interviews. No date. Usability First. Accessed 1 November 2016. http://www.usabilityfirst.com/usability-methods/one-on-one-interviews/

Responsive Web Design: What It Is And How To Use It. 2011. Smashing Magazine, 12 January. Accessed 15 September 2016. https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/

Skeuomorphism. No date. Techopedia. Accessed 12 August 2016. https://www.techopedia.com/definition/28955/skeuomorphism

Stojanov, D. 2015. A Visual Guide to CSS3 Flexbox Properties. Scotch.io, 13 April. Accessed 6 June 2016. https://scotch.io/tutorials/a-visual-guide-to-css3-flexbox-properties

The Adobe Photoshop family. No date. Adobe. Accessed 10 August 2016. http://www.adobe.com/products/photoshopfamily.html

The Gestalt Principles. No date. Spokanefalls. Accessed 10 August 2016. http://graphicdesign.spokanefalls.edu/tutorials/process/gestaltprinciples/gestaltprinc.htm

The Importance of Color in Web Design. No date. Studio 1 design. Accessed 28 September 2016. http://studio1design.com/the-importance-of-color-in-web-design/

Treder, M. 2016. Wire framing, Prototyping, Mocking - What's the Difference? Designmodo, 8 September. Accessed 2 November 2016. https://designmodo.com/wireframing-prototyping-mockuping/

Usability Testing. No date. Usability First. Accessed 1 November 2016. http://www.usabilityfirst.com/usability-methods/usability-testing/

Vendor Prefix. No date. Mozilla Developer Network. Accessed 1 September 2016. https://developer.mozilla.org/en-US/docs/Glossary/Vendor_Prefix

Visual vs. Mixing color wheel. No date. Color-wheel-pro. Accessed 24 August 2016. http://www.color-wheel-pro.com/color-wheel-types.html

Weevers, T. 2012. A Cry For Looking To Other Methods For User Centered Design. Usabilla, 24 April. Accessed 11 October 2016. http://blog.usabilla.com/a-cry-for-looking-to-other-methods-for-user-centered-design/

Wireframe. No date. Webopedia. Accessed 25 August 2016. http://www.webopedia.com/TERM/W/wireframe.html

What is wireframing? No date. ExperienceUX. Accessed 1 November 2016. http://www.experienceux.co.uk/faqs/what-is-wireframing/

Wireframe. No date. Webopedia. Accessed 25 August 2016. http://www.webopedia.com/TERM/W/wireframe.html

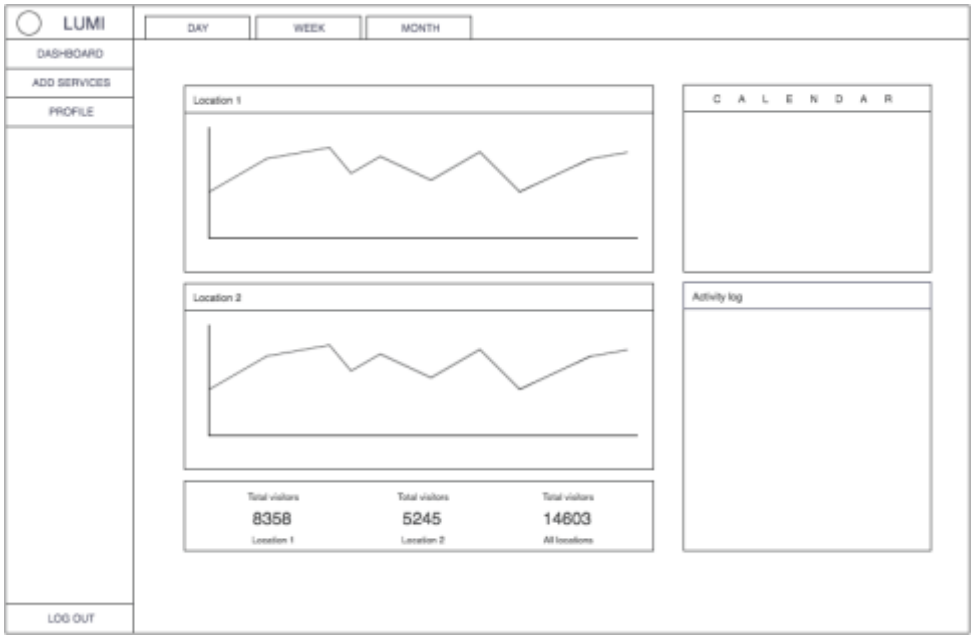
Illustrations

Figures

Appendixes

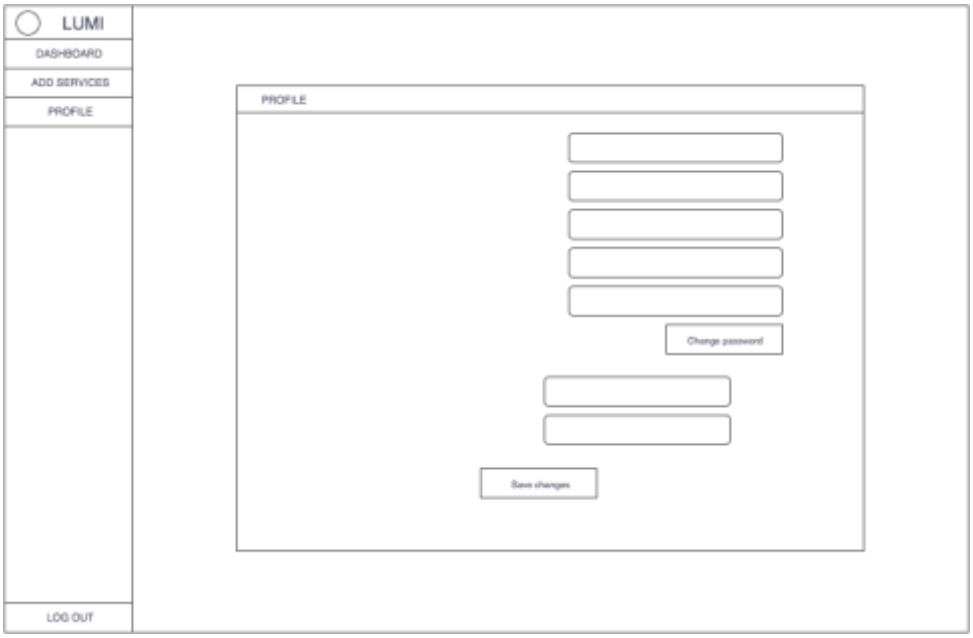
Appendix 1: First wireframes

Login page wireframe



Dashboard page wireframe

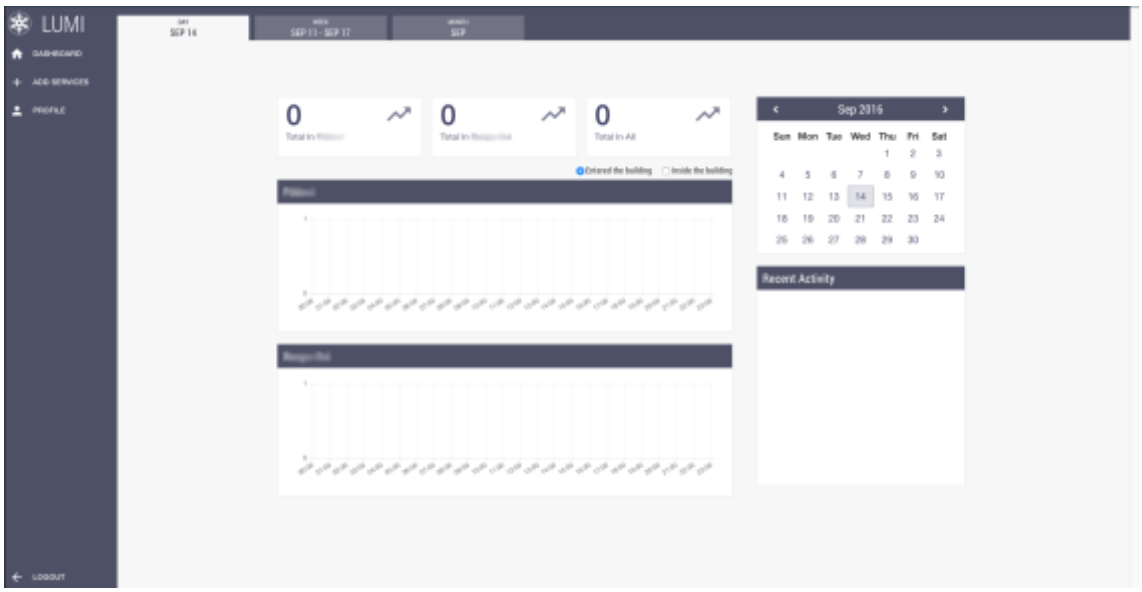
''Add'' popup field wireframe



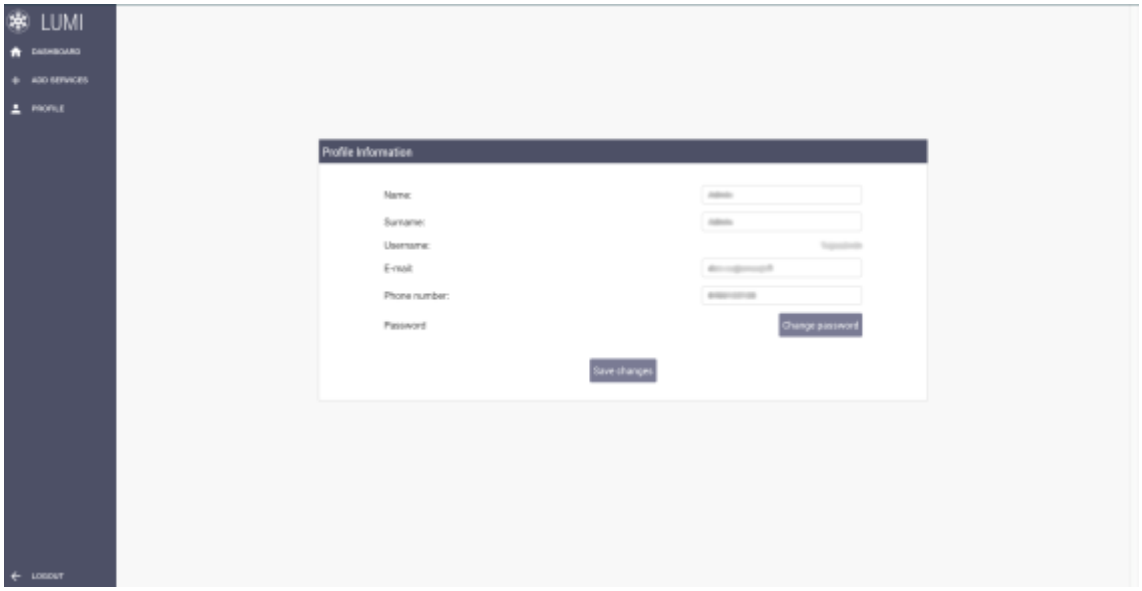
Profile page wireframe

Appendix 2: First mockups

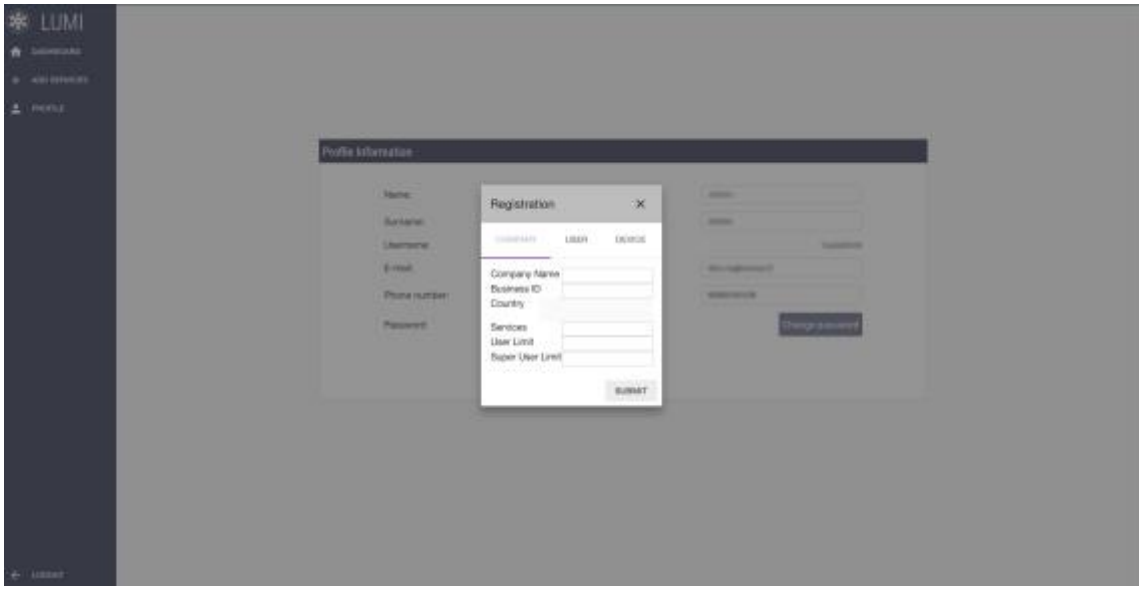
Login page



Dashboard page



Profile page



“Add” popup

Appendix 3: Interview with user testing environment representative

User feedback (31.08.2016)

Did you find the application easy to understand and navigate and did the layout made it easy to find the information that you were looking for?

> The sidebar navigation was clear and simple. The dashboard could've be more customizable e.g. able to select default device or at least quickly navigate to see their data.
>
> Otherwise viewing current and past data in dashboard was simple, essential information was displayed efficiently.

Was it easy to find out which elements were functional and which weren't? If there were any difficulties, which elements caused them?

> Some of the sidemenu buttons were either disabled or somehow broken. Clicking icon did not work whereas clicking text next to them worked with some buttons.
>
> Otherwise color scheme made it clear in most cases which elements are and are not clickable.

Was the colour scheme of the application enjoyable and did it make it easy to see all the information?

> Overall yes.

What seemed unnecessary?

> Selection between currently inside and number of people coming in was seemingly not working. Currently inside kept showing numbers between 0-5 while people coming in was in hundreds

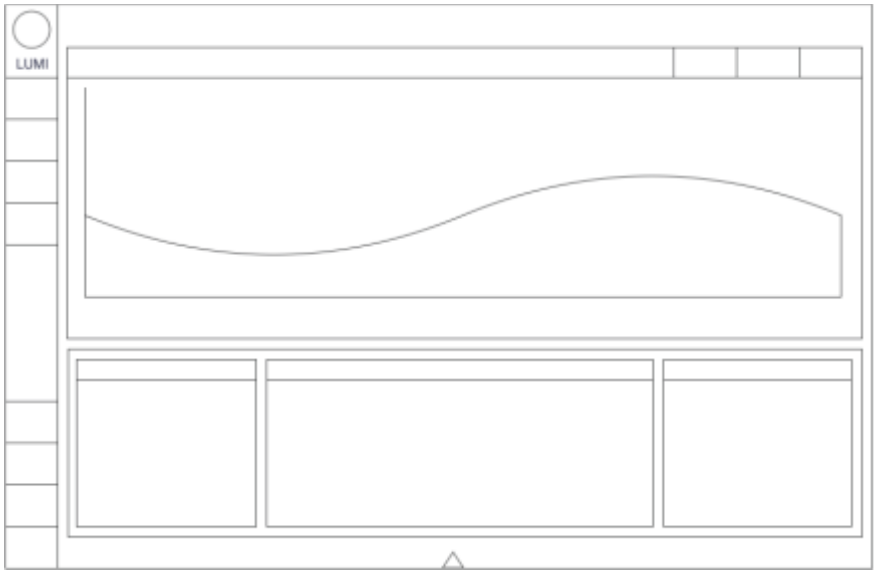
What else would you like to see in the application in the future, if you chose to continue using it?

> Ability to create different kinds of reports, more customizable.
> **Support multiple screen sizes**, currently had problems viewing site with laptop.

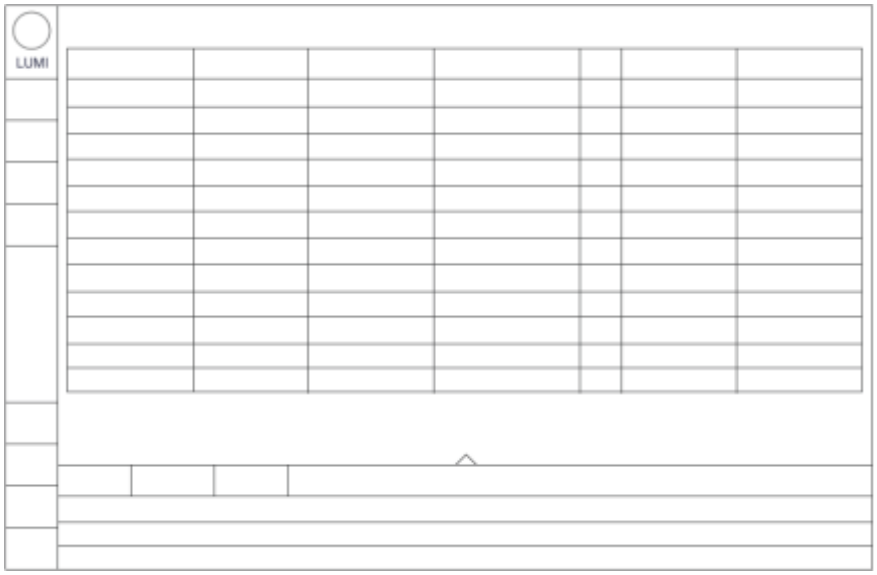
Anything else that you'd like to add?

> Help section with basic guides how to use reporting tool.

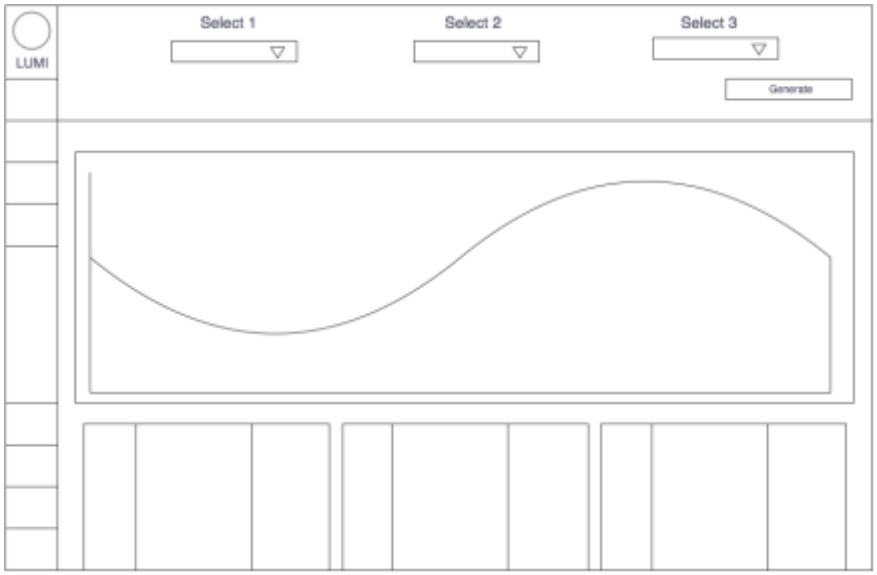
Appendix 4: Improved wireframes



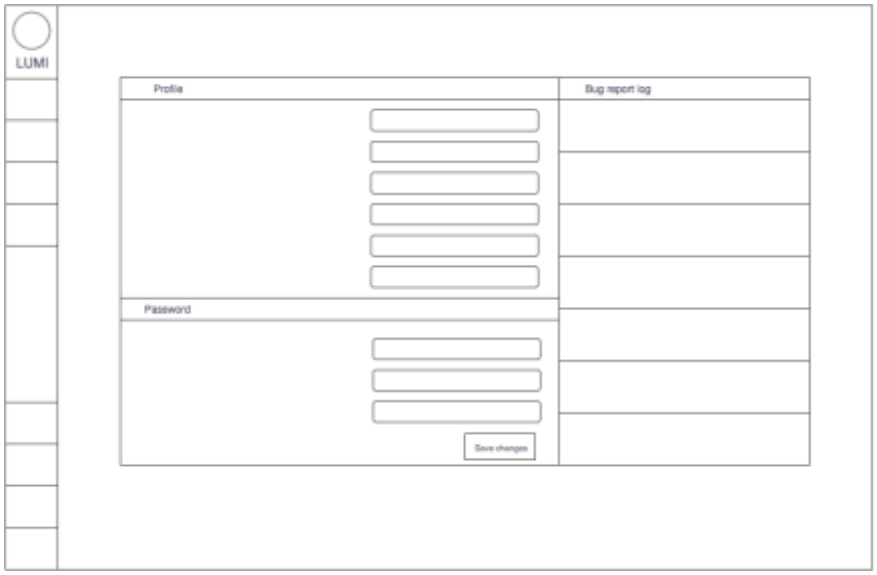
Dashboard page wireframe



Control page wireframe



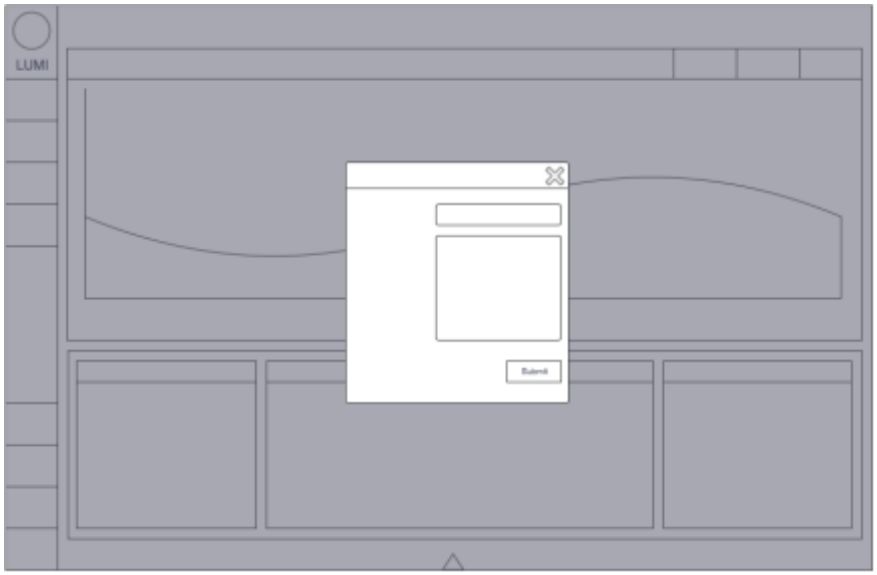
Report page wireframe



Profile page wireframe after the changes

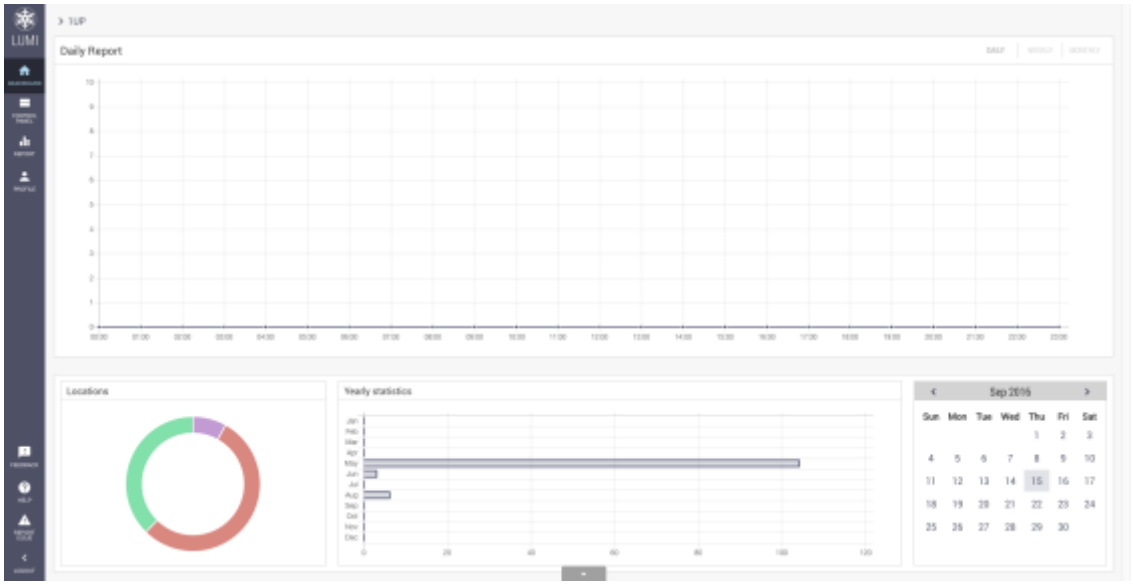
Feedback page wireframe

FAQ page

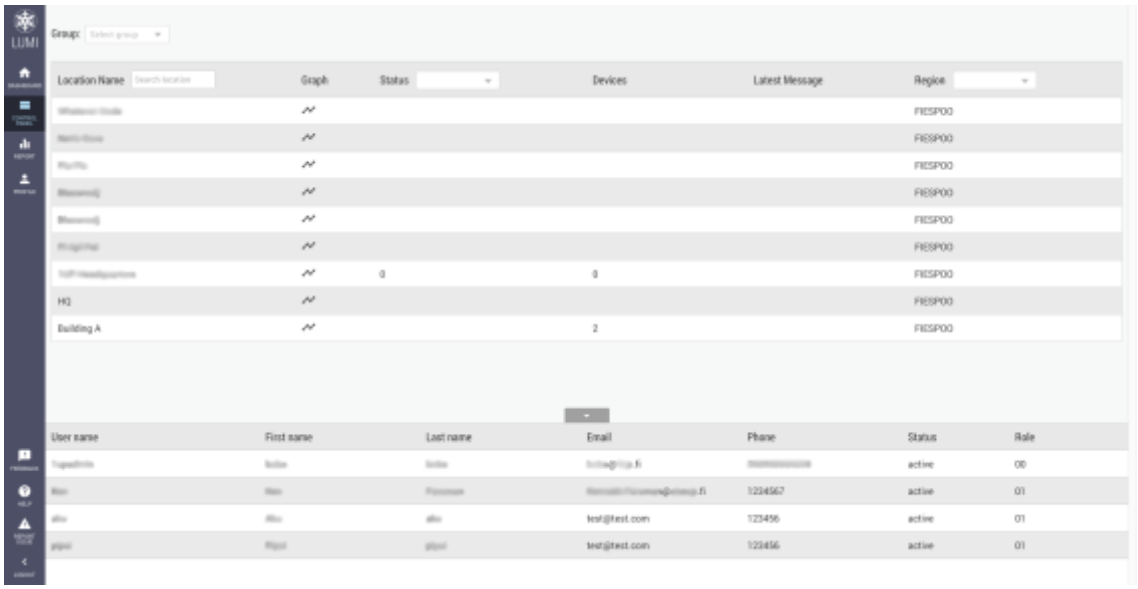


Popup wireframe for bug report and add options

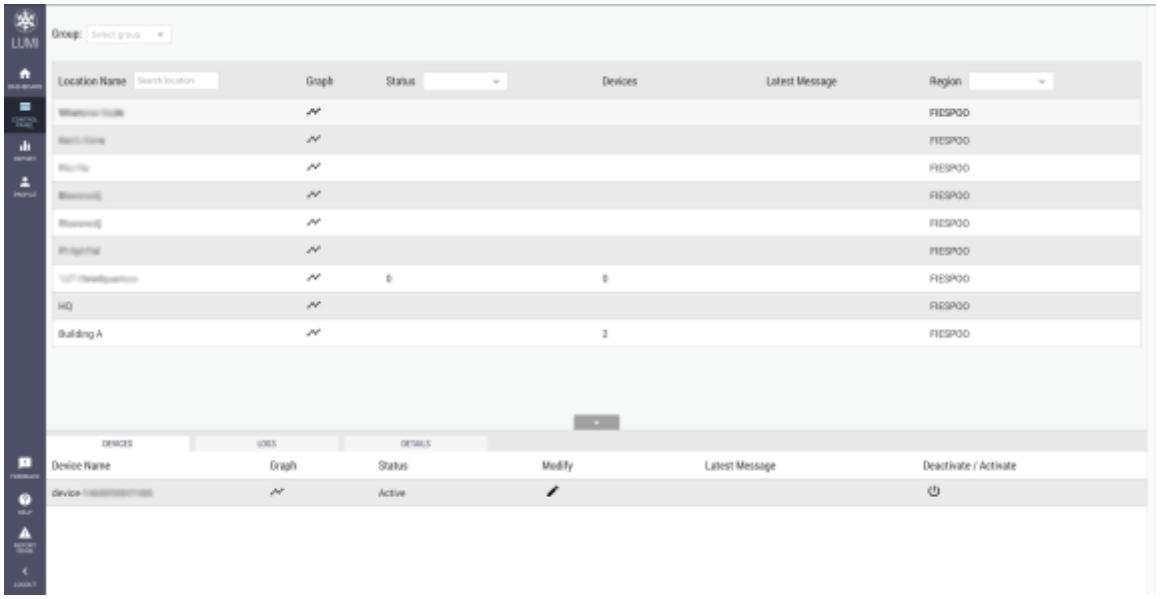
Appendix 5: Improved mockups



The new dashboard design

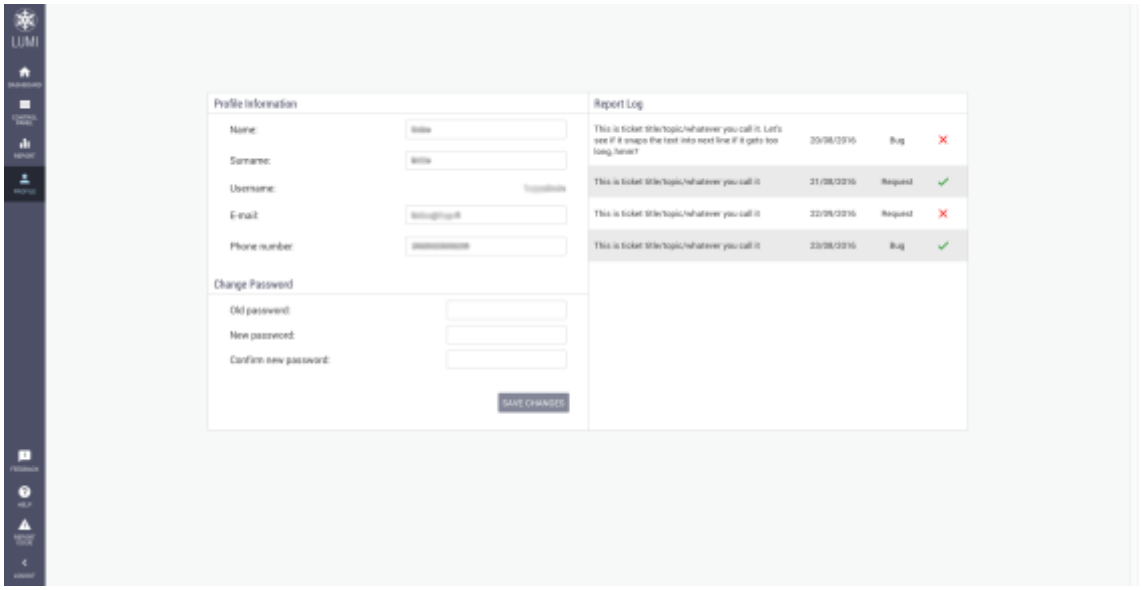


The control panel with the dropup displaying users

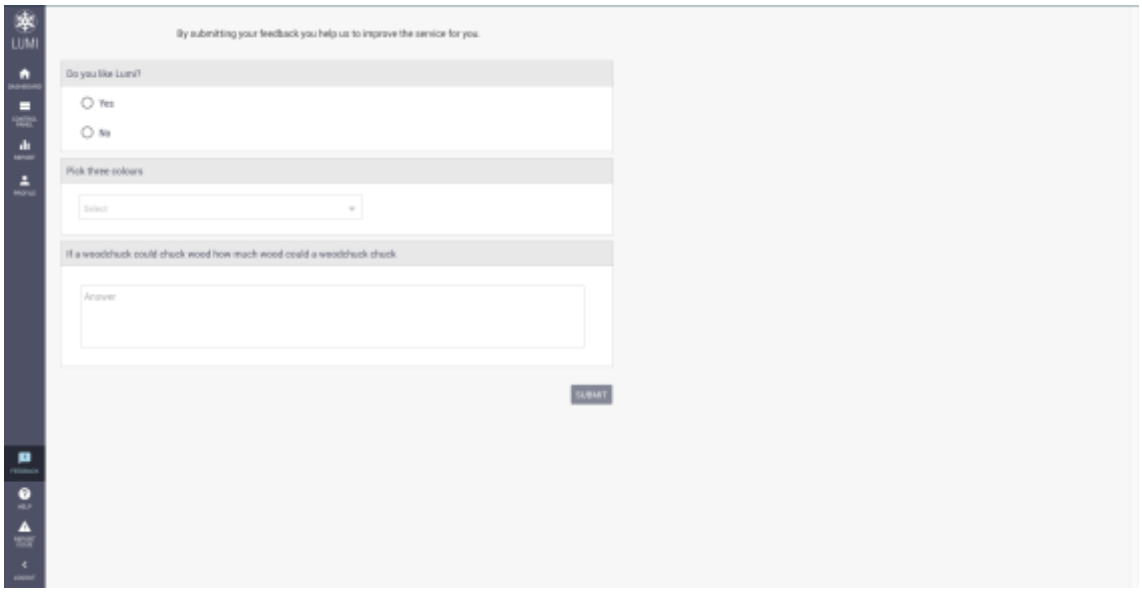


The control panel with the dropup displaying devices

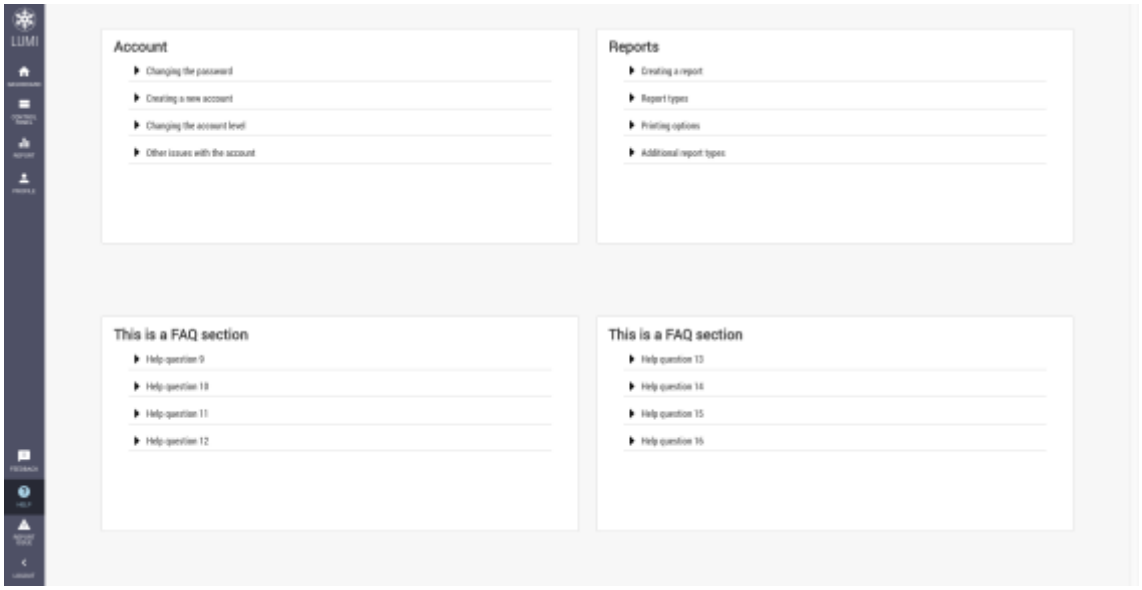
The report generator



The profile page with the bug report log



User feedback page



FAQ page

Bug report popup